Artificial intelligence techniques in flood forecasting

by

Gustavo Cerda-Villafana, B.Eng., M.Eng.

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Engineering, Department of Civil Engineering.

January 2005

Approximately 48,000 words

Abstract

The need for reliable, easy to set up and operate, hydrological forecasting systems is an appealing challenge to researchers working in the area of flood risk management. Currently, advancements in computing technology have provided water engineering with powerful tools in modelling hydrological processes, among them, Artificial Neural Networks (ANN) and genetic algorithms (GA). These have been applied in many case studies with different level of success. Despite the large amount of work published in this field so far, it is still a challenge to use ANN models reliably in a real-time operational situation. This thesis is set to explore new ways in improving the accuracy and reliability of ANN in hydrological modelling. The study is divided into four areas: signal preprocessing, integrated GA, schematic application of weather radar data, and multiple input in flow routing.

In signal preprocessing, digital filters were adopted to process the raw rainfall data before they are fed into ANN models. This novel technique demonstrated that significant improvement in modelling could be achieved. A GA, besides finding the best parameters of the ANN architecture, defined the moving average values for previous rainfall and flow data used as one of the inputs to the model. A distributed scheme was implemented to construct the model exploiting radar rainfall data. The results from weather radar rainfall were not as good as the results from raingauge estimations which were used for comparison. Multiple input has been carried out modelling a river junction with excellent results and an extraction pump with results not so promising.

Two conceptual models for flow routing modelling and a transfer function model for rainfall-runoff modelling have been used to compare the ANN model's performance, which was close to the estimations generated by the conceptual models and better than the transfer function model.

The flood forecasting system implemented in East Anglia by the Environment Agency, and the NERC HYREX project have been the main data sources to test the model.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Ian David Cluckie, and Dr Dawei Han, for their invaluable help throughout the research project described in this thesis.

I would also like to thank to all the present and past members of the Water and Environmental Management Research Centre (WEMRC), who not only provided help whenever I needed it, but also made more fulfilling my time at Bristol. I would like to mention particularly to Dr Miguel Angel Rico-Ramirez and Efren Gonzalez-Ramirez for providing data processed by their software programs and Dr David Hamish Harvey for his clever comments and broad help. Immeasurable thanks to the secretary of the WEMRC, Marghi Peacock.

I acknowledge The Environment Agency, Anglian Region, in particular to Dr. David Price, for providing the Welland and Glen Model data.

I acknowledge all the people who participated and collected the raingauge, flow gauge, and radar data during the HYREX project.

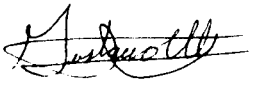
I acknowledge the DHI Water & Environment for providing the software package MIKE 11, and the U.S. Army Corps of Engineers for their free software system HEC-RAS.

Living in Bristol for more than four years has been a rewarding experience thanks to all the people I met and am grateful to, but I want to mention particularly to Elena, Enza, Osiris, Rodolfo, Rodrigo, Elvira, Flor, Sergio and Martina.

This research work has been possible thanks to the National Research Council of Science and Technology (COnsejo NAcional de Ciencia Y Tecnología - CONACYT), who provided my funding under the scholarship 138597.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original except where indicated by special reference in the text and no part of the dissertation has been submitted for any other degree. Any views expressed in the dissertation are those of the author and in no way represent those of the University of Bristol. The dissertation has not been presented to any other University for examination either in the United Kingdom or overseas.

SIGNED:  DATE: 12/JAN/2005

Gustavo Cerda-Villafana

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

All the earth is a grave and nothing escapes it,
nothing is so perfect that it does not descend to its tomb.
Rivers, rivulets, fountains and waters flow,
but never return to their joyful beginnings;
anxiously they hasten on the vast realms of the rain god.
As they widen their banks, they also fashion the sad urn of their burial.

Nezahualcoyotl, Aztec King (1431-72)

Our planet is a place experiencing continuous changes, but these changes are not at random or necessarily extremely complex. They are subjected to certain patterns or rules and the observance and study of these rules (the so called 'laws of nature ') allows the existence of Science. Hence, Science is based on the principle of repeatability: each time a system undergoes similar conditions -both internal to the system and as exerted externally on the system- it is expected the system exhibit a similar response. Prediction and forecasting exploits this principle by using the

observed behaviour of a system to predict a future behaviour when similar conditions recur.

A river catchment is one such system; when external elements like rainfall, heat transfer, and radiation combine with internal elements like state of vegetation, soil moisture, and groundwater levels, the system generates runoff in equivalent quantities whenever those elements present the same conditions. The need to forecast floods and droughts, managing and planning water availability and analysing possible scenarios before modifications to a hydrological system can be addressed is the basis of system simulation.

Different approaches to modelling hydrological systems have been proposed including the early efforts of Pierre Perrault in the 1970s (Ward and Robinson, 1990). Within those models developed for flood forecasting most of them serve as research tools but, many others are used in operational flood forecasting systems. These models present the potential user with a broad choice in basic structure, degree of complexity, data requirements, and other characteristics (WMO, 1992). Nevertheless, the need for easily built, maintained, and updatable hydrological models is still in demand as is inferred from the models currently proposed in the literature.

In this thesis a model that takes advantage of present computational power is developed to forecast runoff and flow based on previous rainfall and flow events, that is, based on the principle of repeatability, the model estimates possible flow values according to rainfall values and flow values downstream from flow values upstream in a river or channel.

Before presenting the model, in *Chapter two* there is an introduction to hydrological models and to artificial intelligence techniques. In the former a deductive process is observed: it starts presenting the hydrological cycle and concludes with the classification of hydrological models. In the latter a brief description of the most common techniques considered within the scope of artificial intelligence is pre-

sented and, for those which have been applied to hydrological simulations a review of related papers is included.

The core of the model is an *artificial neural network* and its mathematical principles are described in *Chapter three.* A serious challenge for any model based on artificial neural networks simulating rainfall-runoff relationships is the high variability present in rainfall estimations compared with the smooth profile generated by the runoff. Although an artificial neural network acts like a filter, it is not efficient enough, as the analysis in section 6.8 shows; the simulated flow can often have higher variability than the observed flow. Therefore, the application of a filter to the rainfall data, together with the inclusion of other preprocessing methods have been a determinant factor for improving the model's performance. Applying a filter to a time series (like a rainfall series) shifts the signal back in time an amount related to the filter's order. In the Brue catchment, which was the catchment under study in the NERC HYREX project (Moore et al., 2000), the variation in efficiency between two different forecasting times (or lead times) is shown to demonstrate the trade-off between lead-time and performance. A Transfer-Function model which, likewise, uses rainfall and runoff data is implemented for comparison.

A network of artificial neurons can be formed by an infinite number of combinations; therefore the selection of the most adequate architecture is an element of concern for a modeller working with this tool. The option selected in this thesis to deal with this problem is the application of a genetic algorithm, which is a robust tool employed in parameter estimation and it is described in *Chapter four.* Genetic algorithms already have been applied to the selection of the optimal architecture of artificial neural networks (see Abrahart et al. (1999); Petridis et al. (1998)) but, besides this objective, in this thesis the genetic algorithm has been applied also to the selection of the extent of a moving rainfall average.

Weather radars play a crucial role in quantitative rainfall measurement over extended and sometimes difficult to access areas. Although they still face serious

challenges to generate continuous reliable estimations (Harrison et al., 2000; Collier, 1996), they have experienced a continuous evolution, from not being considered as a meteorological tool by the Director-General of the Meteorological Office in its 1960 book (Sutton, 1961) or just considered for tracking storms or locating radiosonde balloons (Petterssen, 1958), to play an active part of the Meteorological Office's forecasts. For this reason the last part of this thesis studies the model's response to radar rainfall estimations.

Radar rainfall data used for this analysis was generated within the HYREX project. The Brue catchment was scanned by three radars located at Cobbacombe, Wardon Hill, and Chilbolton. The radar data set has been implemented as a 5 km square grid, in cartesian coordinates, covering this catchment. With these distributed rainfall data three different implementations have been studied. These comprise *average values* which are the average of the 5 km squares overlapping the Brue catchment. With this data set the model has been tested like the model implemented in *Chapter three* which has been run with one-dimensional raingauge data. The *matrix and average values* implementation has been developed to test the model with distributed rainfall data at time t and average values for times $t-1, t-2, ..., t-n$. The last implementation, *matrices* makes the model work as a distributed model.

Flow upstream-flow downstream is the relationship simulated by the model for flood forecasting in *Chapter six*. In this case two conceptual models have been used to compare the model and to analyse some of its characteristics. These two models are implemented in two software packages, MIKE 11 and HEC-RAS. To test these two conceptual models data for modelling the Welland and Glen catchment has been used. This catchment, in East Anglia, was modelled using MIKE 11 by the Environment Agency and, at present is in operation for real-time flood forecasting. Calibrated data from this catchment have been used to make the simulations with the three models. Prior to using the artificial neural network model some of the characteristics of the two conceptual models have been compared.

An artificial neural network has inherent limitations, like simulating variables beyond the range with which it was trained and the range of noise it can deal with. These circumstances have been analysed after comparing the model with the two conceptual models. Two other capabilities of an artificial neural network model have been explored: its capacity to simulate a river junction, feeding the model with the two upstream flows and the model simulating the downstream flow. The second has been its capacity to simulate flow extraction from the stream being simulated.

The general objective of *Chapter six* has been to analyse the capacity of the artificial neural network model to simulate a river-network system. The study of its limitations and its capabilities has been used to test the validity of the artificial neural network model.

A discussion of the results is supplied at the end of each chapter and *Chapter 7* then provides conclusions and recommendations for future work.

Chapter 2

Artificial Intelligence in Catchment Hydrology

> Nothing on earth is so weak and yielding as water, but for breaking down the firm and strong it has no equal
>
> Lao-Tsze (4th century BC)

2.1 Introduction

In order to evaluate the artificial intelligence techniques analysed in this work a prior general knowledge of them and, similarly, the field they are applied to, is necessary. This chapter begins with a brief description of the area within hydrology where these techniques are intended to be applied. The second part of the chapter provides a literature review and discussion about the application of artificial intelligence techniques on flood forecasting.

2.2 The hydrological cycle

The molecule of water consists of one oxygen and two hydrogen atoms. Its molecular weight per mole is 18 grams and, like other substances with around the same molecular weight, would be gaseous at standard temperature but, due to the hydrogen bonding between the water molecules, the Earth experiences liquid water on its surface. Because of this characteristic the latent heats of fusion and evaporation and the heat capacity of water are all unusually high. For these reasons water serves as a temperature regulator and lakes and oceans help regulate the climate. Water is widespread all over the world but not in a homogenous way. Oceans concentrate 96.5% of all the superficial water and cover $361 \text{x} 10^6$ km^2 or 71% of Earth's surface. On land, there are extreme contrasts such as the Atacama Desert in Chile, where in some spots rainfall has never been recorded. Others, on the contrary, have intense rainfall throughout the whole year such as the Southern Alps in New Zealand, with an estimated average of 11 m per year! (Wratt et al., 1996). There are also fresh water reservoirs and lakes such as Lake Superior with 82,100 km^2 on the Canada-U.S.A. border, or lake Baikal in Russia, the World's biggest in volume containing 20% of the Earth's fresh water. Water exists in three states: liquid, solid and vapour. The solid state is known as ice and snow and is found in polar and alpine regions. As liquid, water covers the Earth's surface as oceans, seas, lakes, rivers and soil moisture, and beneath the surface as groundwater. Air contains water in its three states: water vapour, water droplets and ice crystals. Water is, as well, a component of different minerals and a basic constituent of the biosphere. Table 2.1 shows ing the values for volume and residence time of water as it is found on the Earth and indicates the importance of each mode of occurrence.

The interdependence and continuous movement of all forms of water provide the basis for the concept of the *hydrological cycle*, see (Fig. 2.1), which has long been a central concept in hydrology (Jones, 1997; Hornberger, 1998; Ward and Robinson, 1990; Dingman, 1994). From all the elements of the hydrological cycle, this thesis

Table 2.1: Size and residence time for major reservoirs in the hydrological cycle

Parameter	Volume (km^3)	Volume(%)	Residence time (yr)
Water in land areas:	47,971,710	3.5	
Fresh lakes	91,000	0.007	4 (all surface water)
Saline lakes	85,400	0.006	
Rivers	2,120	0.0002	
Marshes	11,470	0.0008	
Soil moisture	16,500	0.0012	
Fresh groundwater	10,530,000	0.76	20,000 (all
Saline groundwater	12,870,000	0.93	subsurface water)
Biological water	1,120	0.0001	
Icecaps and glaciers	24,364,100	1.76	
Atmosphere	12,900	0.001	0.02
Oceans	1,338,000,000	96.5	2,650
Total	1,385,984,610	100	

Source: Maidment (1993)

focuses mainly on rainfall and surface runoff. The former, though a complex subject on its own, is used mainly as an input parameter in the models studied here. The latter, though mankind has developed multiple forms of interaction with it, is analysed primarily as flow in rivers and its hazardous extreme flooding events.

2.3 Influence of man on runoff

Since the first dam was built in ancient Egypt, around 2950-2750 B.C (Smith, 1971), humankind has been influencing surface runoff. Due to the fact that natural processes are diverted, obstructed, delayed or accelerated, the effect of human activity on catchments must be, in general, taken into account, independently of the modelling approach, in any runoff analysis.

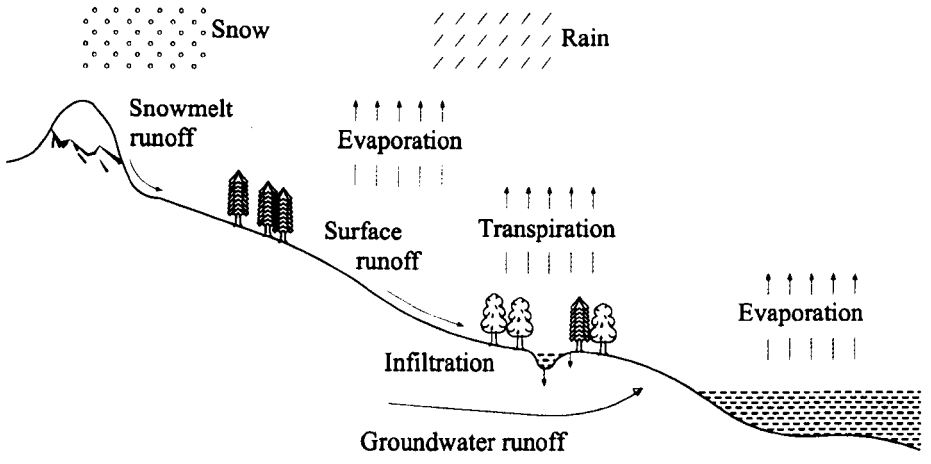


Figure 2.1: Hydrological Cycle.

Though tilling is been done since ancient times, it has had minor influence, or at least not properly documented, on total catchment runoff. It is not until mechanisation and new technology was developed during the 18th century (Federico, 2000) that extended land and water use began to significantly affect the natural catchment system. Change in land use by deforestation, reclamation of wetlands, forestry, etc., has an impact that varies according to the geography and percentage of the affected catchment, soil and plant type, seasonality of the changes, etc. (Bosch and Hewlett, 1982; Lorup et al., 1998; Van Dijk and Bruijnzeel, 2001), e.g.: deforestation is said to cause an increase in annual mean discharge at small scale (<1 km^2) but in land cover in tropical meso- or large-scale river catchments (>100 km^2) usually there are not similar relationships (Costa et al., 2003). Thus, in order to integrate the impact of agricultural activity on a catchment analysis, it is necessary to use information available such as historical rainfall-runoff and land use data for the catchment under study using established values for some of the elements involved such as water

pumped for irrigation, infiltration change due to crop growth, plant evapotranspiration, drainage from crop fields, etc. Any of these data can have an impact on the process and therefore, its inclusion may vary according to availability and the model used.

The impact of hydraulic structures (dams, weirs, bridges, sluices, etc.) on runoff and stream flow is widely addressed and will not be described here. Likewise, artificial channel modifications such as enlargement, straightening, construction of relief and bypass channels and river diversions, are all well documented in literature. These elements have been considered as factors affecting the model developed in this thesis.

Whenever it is addressed, urbanization presents a complex challenge to any hydrological analysis. This element is out of the scope of this thesis. All the analyses have been performed on rural areas.

2.4 Rainfall-runoff relationship

Rainfall-runoff relationship is a process that occurs in both spatial and time domains. Spatial domain is defined by catchment boundaries, rainfall distribution (grid system for radar or satellite rainfall estimation; Thiessen polygon, isohyetal or other mathematical methods for rain gauge estimation), topography conditions, soil layers, stream network, etc., all that has any tangible effect on any resulting estimation. The time domain is defined by the variation of the estimate for each discrete rainfall point along time and on runoff conditions, variation on surface, subsurface and groundwater flows being treated likewise.

Rainfall series present seasonal variation that must be taken into account; Fig.2.2 shows an example of such variation where annual variation can be differentiated,

roughly, between wet and dry months. Even so, rainfall series are said to be stochastic, in that exact predictions are impossible. Uncertainty[1] in any possible rainfall prediction would be based on both time scale and lead time. The larger the time scale (from minutes to hours, days, months and years) the smaller the uncertainty whilst the longer the lead time (1, 2, 3 hours, days or months) the larger the uncertainty.

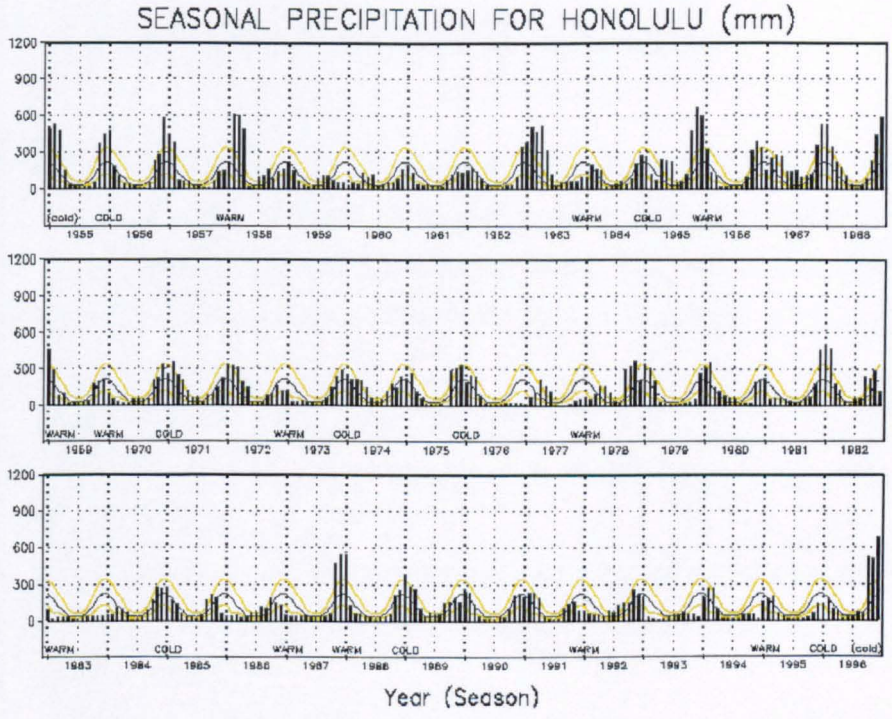


Figure 2.2: Honolulu Monthly Rainfall for years 1955-96. Source NOAA, 2004

On the other hand, runoff series can be predicted with higher levels of accuracy. If a time series can be predicted exactly, it is said to be deterministic. In the case of runoff series, these can be deterministic up to certain limit where flow is simulated using as input rainfall and maybe other variables such as temperature, groundwater extraction, melting snow, etc. These runoff predictions are, understandably, subject

[1]Understanding uncertainty as the estimation of potential error

to variable levels of uncertainty.

Rainfall-runoff relationships are implemented in models with different levels of complexity and purpose. The next section presents a definition and classification of models in order to classify the model developed in this thesis.

2.5 Hydrological models classification

The aim of this thesis is the development of a hydrological model based on artificial intelligence techniques. The core of the model will be based on Artificial Neural Networks and assisted by Genetic Algorithms. Hence, from now on, the model will be referred to as Artificial Neural Network model or ANN model.

A logical sequence in classifying hydrological models is starting with a proper definition:

> Hydrological models are abstract representations of hydrological systems relating something unknown (the output) to something known (the input).

In the case of the ANN model, the known input is precipitation and the unknown output is runoff, or the known input is upstream flow and the unknown output is downstream flow.

Models take a variety of forms. *Physical models* are reduced-dimension representations of real world systems. *Mathematical models* on the other hand, consist of mathematical relationships that represent the response of a hydrological system to a change in hydrometeorological conditions.

The mathematical models can be aimed for long-term runoff forecasting, like the proposed ANN model, in which case they are said to be continuous, or for single events, like the duration of a storm, in which case they are said to be event based models.

A *distributed model* is one in which the spatial (geographic) variations of features and processes are considered explicitly, while in a *lumped model*, these spatial variations are averaged or ignored (HECHMS, 2000). The ANN model takes mainly the second approach but is tested as a distributed model when evaluating weather rainfall data (described in Chap. 5).

According to the knowledge base upon which a mathematical model is built it is classified as a *conceptual model* or an *empirical model*. The former is built upon a base of knowledge of the pertinent physical, chemical, and/or biological processes that act on the input to produce the output. The latter is built upon observation of input and output, without seeking to represent explicitly the process of conversion. An Artificial Neural Network generates its output estimations after being trained using only the required data series, hence, the ANN model fits in the latter classification. It can generally be described as a data-driven modelling approach.

The last model classification is related to their linearity. A model is *linear* if it satisfies the principle of superposition. It means, in mathematical terms:

$$f(a \cdot x_1(t) + b \cdot x_2(t)) = a \cdot f(x_1(t)) + b \cdot f(x_2(t)) \tag{2.1}$$

where a and b are constant real numbers, and x_1 and x_2 are any arbitrary functions of an independent variable t. This equation expresses the two inherent properties of any linear system:

1. *homogeneity* the output produced by a constant multiplied by the input is equal to the constant multiplied by the output of the input alone.

2. *additivity* the output of a sum of inputs is equal to the sum of the outputs produced by each input individually.

A system is non-linear if it fails to satisfy either or both of these conditions. ANNs are non-linear systems and hence, adequate to model hydrological processes which are considered non-linear (Coulthard et al., 1997).

2.6 Real time management

Population growth and industrialising economies all over the world are demanding more and more from limited natural resources. Water is one of the resources being seriously threatened by overexploitation and/or contamination in much of the world, see, as an example, the projection of water scarcity for 2025 shown in Fig. 2.3. Hence, there is an increasing pressure on water management on several topics:

- Water supply for human consumption.
- Efficient water use for agricultural activities.
- Adequate treatment of sewage and wastewater.
- Ecosystem protection.
- Water quality and leakage control for urban systems.
- Recreational accessibility wherever is available.
- Flood management and control.
- Ground water allocation and permits.

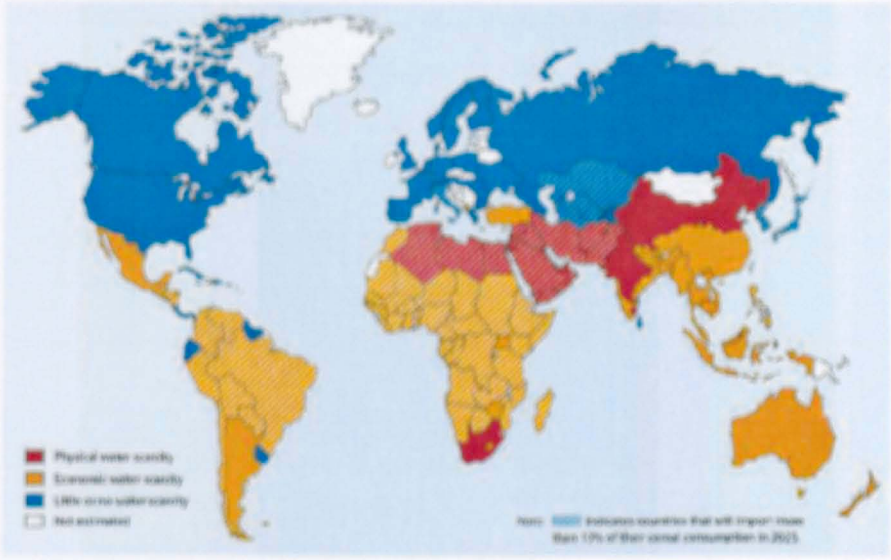


Figure 2.3: Projected Water Scarcity in 2025. Source IWMI, 2004

- Development, operation and maintenance of reservoir systems.

The nature of these topics implies an interconnection among most of them. Working on one specific topic demands understanding and knowledge from several others to determine their interactions. The subject of this work is focused on *Flood Management and Control* and, as such, takes into account the following issues:

- *Water Supply.* When it comes to provide water for human use, it is usually counted as a regular outflow from the system under analysis.
- *Flood Forecasting.* It is requested, for flood assessment, low levels of uncertainty on flow and time of events.
- *Flow Structure Control* As indicated in previous sections, human infrastructure must be taken into account to estimate runoff because, as time goes on, even small rivers and catchments are influenced by.

- *Riparian vegetation* Though this kind of vegetation has little impact directly on a stream under normal conditions, its life cycle can affect the rainfall-runoff relationship and be a major factor under flood conditions.

- *Aquatic vegetation* This is a factor that, most of the times can be considered employing an annual cycle. Nevertheless sometimes it could severely affect a stream by growing up abnormally bulky or, sometimes, usually by pollution, vanish completely.

Real-time management of a catchment involves the use of telecommunication to get real-time data. Microwaves, telephone lines, internet and optic fibre are the media through which information from raingauges, current meters, radars and even satellites is transmitted.

It involves management of reservoirs and regulating structures and is governed according to specific policies: guaranteeing water supply for human consumption, land and/or industrial use, flood control, ecologic sustenance, hydropower and transport. Managing one, several or all this objectives can be accomplished thanks to software programs that can generate estimations for requested variables almost immediately after receiving updated information.

In this thesis, besides the model developed using artificial intelligence techniques, two software packages used in water management have been employed for river flow simulation. They both can be adopted to work in real-time conditions. One of them, MIKE 11 from DHI Water & Environment is a commercial package and, the other, HEC-RAS from U.S. Army Corps of Engineers is a free option though, if technical support is required, it is obtained through vendors by a fee. There are, of course, many more software packages available in the water management community; Wurbs (1995) describes, though now out of date, many water management models, their capabilities and practical applications.

The ANN model, as said previously, is based on two Artificial Intelligence techniques, hence following is a description of Artificial Intelligence and its most important fields. A review of publications describing applications in hydrology is presented in the fields involved.

2.7 What is artificial intelligence?

The usual introduction to any Artificial Intelligence (AI) book starts with a definition of *Artificial Intelligence* and, though there is not a generally accepted one, all definitions, like the following list, comprise the generation of intelligent behaviour by man-made objects:

- *Artificial intelligence is the enterprise of constructing an intelligent artifact* (Ginsberg, 1993)
- *Artificial intelligence may be defined as the branch of computer science that is concerned with the automation of intelligent behaviour* (Luger, 1998)
- *Artificial intelligence is the study of the computations that make it possible to perceive, reason, and act* (Winston, 1992)
- *Artificial intelligence, broadly (and somewhat circularly) defined, is concerned with intelligent behaviour in artifacts* (Nilsson, 1998)

Developments on AI have been done in two directions, those specifically dedicated to simulate human behaviour and those dedicated to use AI tools to specific tasks (forecasting share prices or weather, software games, odour, noise, and voice detection, etc.). The former are those developments intended to or based on research on human behaviour, brain and senses physiology, computer potentialities, and even some branches of philosophy (Ashby, 1960; Dreyfus, 1992; Williamson,

2004). The latter are focused on applying AI technologies to surpass human abilities in performing certain tasks or automating those human activities when they are repetitive or tedious. Among the activities AI techniques are assigned to perform are the following:

- *Searching*: finding 'good' material after having been provided with only limited direction, especially from a large quantity of available data.
- *Surmounting constraints*: finding ways that something will fit into a confined space, taking apart or building a complex object, or moving through a difficult maze.
- *Recognising patterns*: finding items with similar characteristics, or identifying an entity when not all its characteristics are stated or available.
- *Making logical inferences*: drawing conclusions based upon understood reasoning methods such as deduction and induction.

AI technologies comprise several areas that will be examined in some depth here, but it is first important to understand that the importance of these individual areas has changed over the last two decades. These changes have been based upon the progress in each area, and the needs that each area meets. For example in the early 1980's robotics was a large thrust in artificial intelligence. At that time benefits could be seen in manufacturing applications. In the late 1990's the blossoming of the Internet pushed the importance of intelligent agents forward for performing routine tasks and complex searches (Cooley et al., 1997). At the same time, throughout the 1980's and 1990's, orders of magnitude advances in computer processing power have allowed hurdles in speech recognition and image processing to be overcome.

The following sections include descriptions of several artificial intelligence techniques and specifically those employed in hydrological modelling. A review of related literature is included for comparative purposes.

2.8 Expert systems

Expert systems are designed to enhance the quality and availability of knowledge required by decision makers in a wide range of industries. They augment conventional programs such as databases, word processors, and spreadsheet analysis. Expert systems are characterised by the following points:

- The expert system shell, or interpreter.
- The existence of a "knowledge base", or system of related concepts that enable the computer to approximate human judgment.

The Expert System Shell While any conventional programming language can be used to build a knowledge base, the expert system shell simplifies the process of creating a knowledge base. It is the shell that actually processes the information entered by a user; relates it to the concepts contained in the knowledge base; and provides an assessment or solution for a particular problem. Thus, an expert system shell provides a layer between the user interface and computer operating system to manage the input and output of data. It also manipulates the information provided by the user in conjunction with the knowledge base to arrive at a particular conclusion. The shell manages the user interface, performing functions that range from the validation of numeric values entered on the screen to management of the mouse and the representation of graphical objects.

The Knowledge Base The main purpose of the knowledge base is to provide the guts of the expert system–the connections between ideas, concepts, and statistical probabilities that allow the reasoning part of the system to perform an accurate evaluation of a potential problem. Knowledge bases are traditionally described as large systems of "if then" statements, but this description is misleading because knowledge bases may not contain definitive rules at all, but may contain only associative

relationships among different concepts, statistical information about the probability of certain solutions, or simply large databases of facts that can be compared to one another based on simple conventions intrinsic to the expert system.

The purpose of the expert system is to enhance judgment on the part of the user, not to replace human judgment altogether. Expert systems can provide a relatively inexperienced user with a lucid assessment of a problem where an expert is unavailable. Expert Systems are established for processes where there is a need;

1. for a narrow area of expertise to be more widely known, or
2. to allow sophisticated processes to be run without human intervention.

From the description of expert systems it is clear they are not used in hydrological modelling to generate directly, for example, flow estimations; they are used as an auxiliary tool to assist the modeller/user for building up and/or calibrating models (Lumb et al., 1994; Varas and von Chrismar, 1995). Simanovic (1990) described an expert system for selection of a suitable method for flow measurement in open channels.

2.9 Artificial neural networks

This technology is based loosely upon the cellular structure of the human brain. Cells, or storage locations, and connections between the locations are established in the computer. As in the human brain, connections among the cells are strengthened or weakened based upon their ability to yield "productive" results. The system uses an algorithm to 'learn' from experience. A detailed description of them is presented in Chapter 3.

There is a vast number of publications on artificial neural networks. Maier and Dandy (2000) presents a review of 43 papers dealing with the use of ANNs for the prediction and forecasting[2] of water resources variables. In this article they describe the parameters and architectures presented in each paper and, likewise, the lack in reporting them. The only missing element was to report the input variables each paper was using. From those papers simulating flow as their output variable, only two were forecasting over a length of 5 hours: Golob et al. (1998) estimated 2, 4 and 6-hour lead-times, the last one around the catchment's natural response. They used a neural network with three outputs for generating the three lead times. Regrettably they do not report any normalisation or preprocessing methods. The other paper (Dawson and Wilby, 1998) is reported to have a forecast length of 24 h, but looking at the paper it works with only a lead time of 6 h; it uses storm events from the previous 24 hours as one of the inputs of the model.

Other paper by Maier and Dandy (1998) was intended to forecast salinity but it is an excellent introductory article to artificial neural networks. Sajikumar and Thandaveswara (1999) developed a rainfall-runoff model using monthly data. They described two methods to take account of previous rainfall events and proposing a third one: using finite impulse filters at each input of the neural network. They compared their model with other three "black box" models but do not indicate its forecast length. An interesting approach to flood forecasting is the paper by Gwangseob and Barros (2001). They divided a big catchment (13,071 km^2) in four smaller areas; although satellite, radiosonde, raingauge, and streamflow gauge data are employed, a weather classifier is presented, but specific details on the 4 neural network models used is not provided.

Two excellent references are ASCE (2000b) for an introduction to the basics of artificial neural networks in hydrology, and ASCE (2000a) as a review on their ap-

[2]A **forecast** is a comparatively imprecise statement of the time, place and magnitude of an event whereas a **prediction** is a relatively precise statement giving the time, place, and magnitude of ongoing monitoring results of an event.

plications. These papers, by the ASCE Task Committee on 'Application of Artificial Neural Networks in Hydrology' were open to discussion and give a relevant analysis of the state of development of this tool up to the year 2000. The papers reviewed can be classified in two categories: those using synthetic or simulated data for training and testing the artificial neural networks and those using real data. From this point of view this thesis has taken both approaches: the rainfall-runoff analysis has been performed using observed data and the flow-flow using data from an existing model.

Fernando and Jayawardena (1998) used radial-basis function (RBF) networks for flood forecasting and 1-h step data. The forecasts were only 1-h ahead and the authors compared the RBF networks with a multilayer perceptron (MLP) and both performed well and better than the ARMAX (auto-regressive moving average exogenous) model. Yang et al. (2004) proposed a procedure that integrates a linear transfer function and a self organised map network to determine efficiently the intervals of weights and biases of a flood forecasting neural network.

Relatively few papers describe thoroughly the models used in their studies and even there are papers which leave more questions than answers; Lange (1999) described an artificial neural network for rainfall-runoff modelling and, although the data normalisation and stopping criteria were not mentioned, the author concluded that the number of events to be used for training are the determinant factor for the model's performance without giving number of epochs or number of data for each event. Other work, by Jayawardena and Muttil (2004), presented an adequate model for flow modelling applied to a small catchment (5.22 km^2). They obtained poor results not because of the model itself but because they tried to forecast one day ahead while the catchment's response extends for no more than 3 hours. Any precipitation event would have a minimum effect the next day.

The number of papers presenting applications of artificial neural networks in rainfall-runoff, flow-flow relationships, and hydrological processes in general is vast and keeps growing, demonstrating the importance of these areas and the possibilities

this tool presents.

2.10 Genetic algorithms

To evaluate and locate the best candidates for a task, the Genetic Algorithm (GA) has been found to be a very effective method . GAs borrow from scientific discovery about the evolutionary nature of our genes. They utilize fitness functions, which are relationships among criteria, to grade candidates. They also use evolutionary methods such as crossover and mutation on chromosomes, or strands of information, to find the best examples from a very large field of possibilities. A most complete definition is presented in Chapter 4.

Genetic Algorithms are applied in hydrological modelling as a calibration tool into different schemes like conceptual models (Chérif et al., 2002), derivation of equations (Nandalal, 2002), and tuning the structure and parameters of neural networks (Topping et al., 1998; Abrahart et al., 1999; Yang, 1999).

2.11 Fuzzy logic

Fuzzy logic was introduced by Dr. Lotfi Zadeh in the 1960's as a mean to model the uncertainty of natural language. Fuzzy theory expresses lack of precision allowing a generalization of conventional logic. It provides for terms between "true" and "false" like "almost true" or "partially false".

Dr. Zadeh says that rather than regarding fuzzy theory as a single theory, the process of "fuzzyfication" should be regarded as a methodology to generalize ANY specific theory from a crisp (discrete) to a continuous (fuzzy) form. Thus recently

researchers have also introduced "fuzzy calculus", "fuzzy differential equations", and so on (Bezdek, 1993).

The application of fuzzy logic in hydrology is orientated to system operation or analysis like Mahabir et al. (2003) who present a forecasting model which, based on winter snow levels, forecasts spring runoff in three levels: low, average, and high. Several papers present hybrid models using fuzzy logic and artificial neural networks, like those by Gautam and Holz (2001); Liong et al. (2000) or fuzzy logic and conceptual models (Özelkan and Duckstein, 2001).

2.12 Support vector machines

Support Vector Machines (SVM) are learning machines founded on *Statistical Learning Theory* that can perform binary classification (pattern recognition) and real valued function approximation (regression estimation) tasks. They perform the structural risk minimisation principle (Evgeniou et al., 2000). Support vector machines non-linearly map their n-dimensional input space into a high dimensional feature space. In this high dimesional feature space a linear classifier is constructed. Two results make this approach successful:

- The generalization ability of this learning machine depends on the VC dimension [3] of the set of functions that the machine implements rather than on the dimensionality of the space. A function that describes the data well and belongs to a set with low VC dimension will generalize well regardless of the dimensionality of the space.
- Construction of the classifier only needs to evaluate an inner product between two vectors of the training data. An explicit mapping into the high dimensional

[3] Vapnik-Chervonenkis dimension h of a space of $\{-1, 1\}$-valued functions, G, is the size of the largest subset of domain points that can be labelled arbitrarily by choosing functions only from G.

feature space is not necessary. In Hilbert space inner products have simple kernel representations and therefore can be more easily evaluated.

Support vector machines are a relatively new tool within artificial intelligence but, due to their characteristics and potential there are already some papers presenting applications to hydrological processes. Sivapgragasam et al. (2001) presents a SVM model using Single Spectral Analysis (SSA) to forecast rainfall and runoff data using 1-h time steps. Han et al. (2002)present a variant to SVM named Relevance Vector Machines and used to model flow.

2.13 Automatic speech recognition (ASR)

This technology takes the sound waves produced by a speech and converts them into text content. The process, made possible by lots of computer memory and fast processors, works like this:

First a continuous voice sound waves, captured by a microphone, are fed into a digital converter. This converter takes many samples (like capturing a snapshot) at a very high rate, e.g. 20,000 times per second.

These samples are compared against a large stored template of sounds which match specific text. The computer then outputs the text which most closely matches the template.

For this process to work, the system must first be trained to recognize any specific voice. It does this by asking the user to speak a series of simple phonemes, or parts of speech. After about 30 minutes of training, the system can then begin to derive complex speech patterns from the simple ones you have provided.

The methods used to derive and select from the large numbers of patterns are quite sophisticated. They borrow from techniques such as Markov Processes, which used probabilities to determine what the most likely next syllable or word may be, or from Neural Networks. The ASR technology would likely be of use to anyone who needs to utilize a computer. However it is most beneficial to those who find it difficult to use a keyboard, such as people suffering from Carpal Tunnel Syndrome or those who need hands free access on a manufacturing line.

2.14 Intelligent agents

Intelligent agents (IA), now often known as 'bots', are software technology that performs difficult or repetitive tasks for a user. Using direct commands or on a scheduled timetable, the IAs execute a provided list of instructions known as a script. The intelligent agent technology typically 'borrows' from capability inherent in other AI techniques, especially in the area of search. The IA capability can then add the "ever diligent" capability provided to us by computer processors that can stay awake and work 24 hours a day 7 days a week. A simple script can check for stories about the stock market and then notifies the user via e-mail when one of interest appears. Scripts that are well crafted can perform actions such as periodically:

- Checking for a stock price to hit a certain level and then executing a buy or sell at that price.
- Checking a web site to see if any new documents have been deposited.

Chapter 3

Rainfall-Runoff modelling with ANN

Artificial Intelligence is no match for natural stupidity

Anonymous

3.1 Introduction

An Artificial Neural Network (ANN) is a data modelling tool that is able to capture and represent complex input/output relationships. It could be understood at first glance as an input/output processing unit. It means that, given an input data set $X(n)$ that could be the pixels of an image, a voice, sound or noise signal, odour or aroma levels in a chemical process, etc., the processing unit, by a series of parallel operations, will generate an output $Y(m)$ related to the input: object identified, noise type, trigger signal, etc., see Fig (3.1).

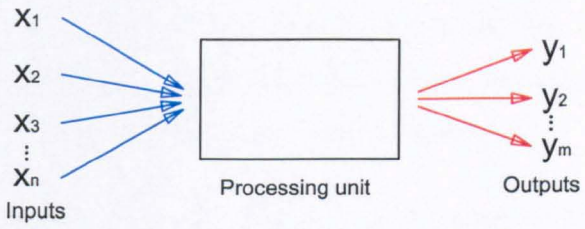


Figure 3.1: Input/output processing unit.

After this first approach, in order to clarify the concept, it could be said that $Y(m)$ is a data set related to $X(n)$ through a transformation process made by the ANN. Before it can generate any useful output signal, the ANN is fed with examples of both input $X(n)$ and output $Y(m)$ data. After a *training* process with such information, the ANN will *learn* to give output values based on known and unknown values of the input $X(n)$.

The first part of the chapter presents the principles and basic theory of the ANNs implemented in this work. Information related to any other model, algorithm or improving method applied in the vast field of ANNs, can be found elsewhere. The chapter continues with the description of an ANN model intended for real-time application to flood forecasting.

3.2 Background

The origin of ANNs was universally established in the paper by McCulloch and Pitts (1943). This pioneering work was followed by the likes by Rochester et al., Minsky, Anderson, Kohonen, Nakano, etc. through the fifties and sixties (Fine, 1999; Haykin, 1999). Developments on the physiological area of biologic neural networks

were a boost for their artificial counterparts; works like Hebb's book *The Organization of Behavior* and Ashby's book *Design for a Brain: The Origin of Adaptive Behavior* were inspirational and some of their principles and ideas translated to the mathematical and computational realms (Haykin, 1999).

A setback to the development of ANNs was the book by Minsky and Papert (1988). The authors demonstrated, using mathematical analysis, that there are fundamental limits to what single-layer perceptrons (a specific type of ANN) can compute. They extended these limitations to multi-layer perceptrons, thereby discouraging the work and support on ANNs during the 1970s. The lack of availability of computers for experimentation contributed likewise to the fact that only a handful of researchers maintained their commitment to ANNs.

The 1980s witnessed a resurgence of interest on ANNs. Hopfield, Rumelhart, McLelland, Barto, Braitenberg, Broomhead, Lowe, Vapnik, Grossberg, Haykin, (Haykin, 1999; ASCE, 2000a; Fine, 1999) are all well-known names on the field and every year new names are added to the list. The application areas range from medicine, marketing and manufacturing to chemical process control and real estate analysis (Bernard et al., 1994).

3.3 Biological neuron

The principle of ANNs is to mimic the function of the brain's neurons, then following is a brief description of a basic cell. Each biological neuron is a processing unit that receives, processes and sends information. Neurons are constituted, roughly, by three parts: *the axon* (transmission line), *the dendrites* (receptive elements) and *the cell body* (called soma), see Fig. (3.2). The axon is the conduit through which the neuron transmits impulses to dendrites of neurons downstream in the signal chain. Dendrites and the axon terminals form the interface where impulses are transmitted

from one neuron to another. The junction between both of them, where the signals travel through, is called *the synapse*. It is at this junction where neurons are excited, inhibited, or modulated. The response is based on the magnitude of the signal and how the synapse has been adjusted to respond through a previous learning experience (Nicholls, 2001).

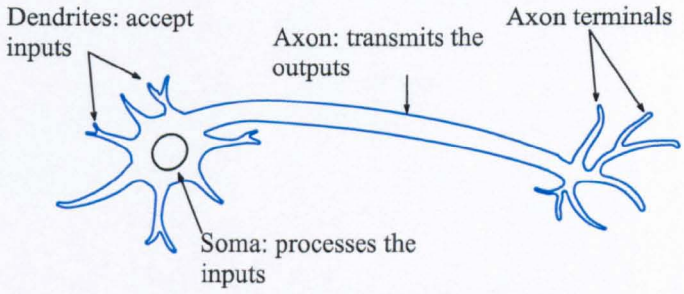


Figure 3.2: Biological neuron structure.

3.4 Artificial neuron model

Based on its biological counterpart, an artificial neuron is an information-processing unit. A model of an individual neuron is shown in Fig. (3.3). A set of connecting links (or synapses) receives the n input signals X_i, multiplying these inputs by the corresponding synaptic weights W_i, $i = 1, 2, 3, ..., n$ (where the knowledge resides). After being weighted, the signals are added up in the summing junction by a *linear combiner*. The activation function $f(\cdot)$ limits the amplitude of the output and gives it its nonlinearity. Its mathematical representation is given in the equations 3.1 and 3.2:

$$\sigma = \sum_{i=1}^{n} X_i W_i \tag{3.1}$$

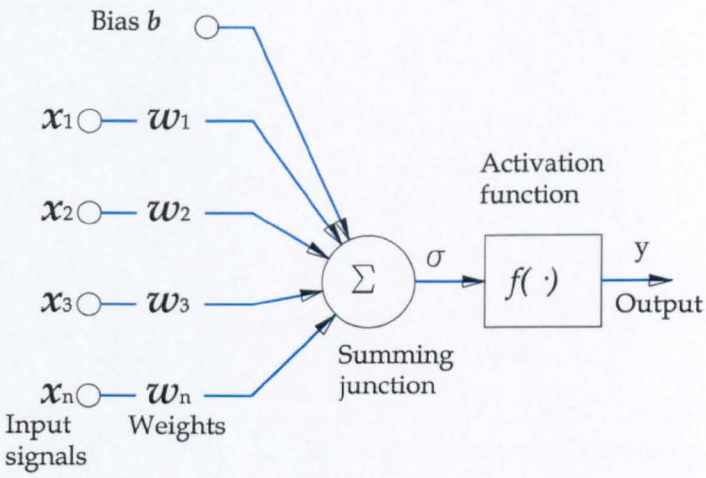


Figure 3.3: Artificial neuron model.

$$Y = f(\sigma + b) \tag{3.2}$$

where b is the *bias*. These components involve certain considerations to be taken into account for an appropriate development of any ANN. In the following sections they are explained in more detail.

3.5 Bias

The *bias* or *offset* is a fixed input and its function could be understood by simplifying Eq. 3.1 to only two elements:

$$\sigma = x_1 w_1 + x_2 w_2 \tag{3.3}$$

This elementary form is able to distinguish between two different categories of values as is shown in Fig. (3.4.a).

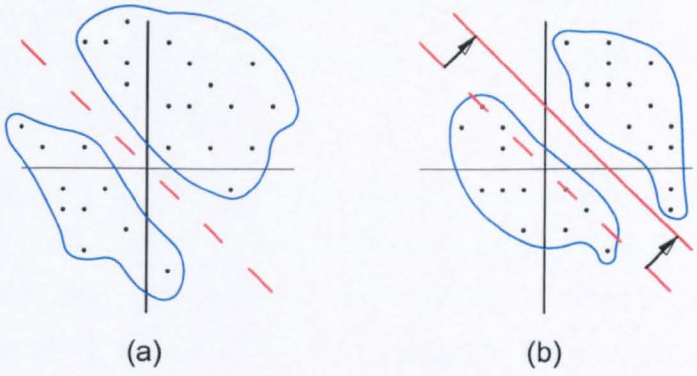


Figure 3.4: (a)Two-categories classification, (b) bias effect.

The division line is known as the decision boundary and its equation is obtained from Eq. 3.3:

$$x_1 w_1 + x_2 w_2 = 0 \tag{3.4}$$

In order to shift the decision boundary from the origin the bias is added to the ANN equation and its graphic representation is shown in Fig. (3.4.b). In more complex systems the bias effect could be more difficult to visualise but the principle remains throughout.

3.6 Weights

Weights are multipliers used in the connections between two neurons. They are dynamically changed during the learning process, storing the acquired "knowledge". Initialisation of weights starts with a random assignment and requires values neither so big as to saturate the activation function, nor so small as to stay around the origin

of the error surface [1]. A proper set of initially randomised weight values have a zero-mean and a variance equal to the reciprocal of the number of synaptic connections of a neuron (Haykin, 1999).

It is widely accepted in the literature that a good weight initialisation is essential to achieve both a fast convergence and avoid saturation (Haykin, 1999; Maier and Dandy, 2000; Fine, 1999). A method of initialisation proposed by Iyer and Rhinehart (1999) specifies a minimum number N of initialisations in order to find the optimal one among them (according to specific conditions) and is based on Eq. 3.5

$$N(t) = \frac{ln(1 - F_W(a))}{ln(1 - F_X(a))}. \tag{3.5}$$

where a is the reference error value or the biggest sum-of-squared errors value to accept, X is the sum-of-squared errors on any individual optimisation, $F_X(a)$ is the fraction of random starts which could result in a value of X less than or equal to "a", W is the lowest value for X, $F_W(a)$ is the fraction of best-of-N X-values that result in a value of W less than or equal to "a".

As an example, in order to get a 90% of confidence ($F_W(a) = 0.90$) that the best-of-N random starts will result in one of the best 10% values for the sum-of-squared errors ($F_X(a) = 0.10$) then $N = (ln(1 - 0.90)/ln(1 - 0.10)) \cong 22$. This result means that the model runs the random weight initialisation 22 times and, through an evaluation process, selects the best one. This value of 22 random initialisations has been utilised for training the models used in this thesis.

[1]Function relating the sum of squared errors over the training sample. It is a performance measure of the system

3.7 Activation function

The activation function of a neuron can be virtually any mathematical function but, in order to: (a) limit the output range, (b) have a continuously differentiable function and, (c) have a computationally low cost first derivative, the most common activation functions used are:

- Threshold Function (also called step function):

$$f(\sigma) = \begin{cases} 1 & \text{if} \sigma \geq 0, \\ 0 & \text{if} \sigma < 0. \end{cases} \tag{3.6}$$

- Linear Function

$$f(\sigma) = \begin{cases} 1 & \text{if } \sigma \geq +\frac{1}{2}, \\ \sigma & \text{if } +\frac{1}{2} > \sigma > -\frac{1}{2}. \\ 0 & \text{if } \sigma \leq -\frac{1}{2}. \end{cases} \tag{3.7}$$

- Sigmoidal Function

$$f(\sigma) = \frac{1}{1 + e^{-a\sigma}} \quad 0 \leq f(\sigma) \leq 1 \tag{3.8}$$

- Hyperbolic Tangent Function

$$f(\sigma) = tanh(\sigma) \quad -1 \leq f(\sigma) \leq 1 \tag{3.9}$$

The graphs for each one of these functions are shown in Fig. (3.5). The first two, threshold and linear functions are not adequate for training using the backpropagation algorithm [2] because the first derivative is zero or constant. The one selected for this work is the hyperbolic tangent function which has the following elementary properties:

[2]This training algorithm is presented in the following sections

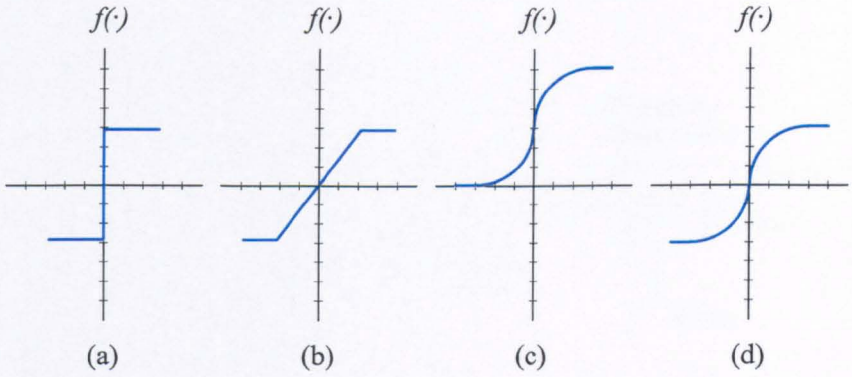


Figure 3.5: Activation functions: (a)threshold, (b)linear, (c)sigmoidal, (d) hyperbolic tangent.

$$\lim_{\sigma \to -\infty} f(\sigma) = -1 \quad \lim_{\sigma \to \infty} f(\sigma) = 1 \tag{3.10}$$

$$\frac{\delta f(\sigma)}{\delta \sigma} = [1 - f(\sigma)][1 + f(\sigma)] \tag{3.11}$$

As is indicated by Eqs. 3.10 and 3.11, the output limits for every neuron using the hyperbolic tangent function are set between 1 and -1. In terms of computational cost, its first derivative is obtained only by two adding and one multiplying operation using the function output value.

Once all the elements of an artificial neuron are formed by the methods described, they are presented in Fig. 3.6 which shows an example of possible values for these elements. The input values x_1, x_2, x_3, and x_4 are multiplied by those of the weights, w_1, w_2, w_3, and w_4. The bias is then added to the sum of these multiplications. The bias is usually initialised to 1 or -1 but this starting value is modified, like those of the weights, by the training process. The total sum is the argument of the hyperbolic tangent function. The result of the function is the neuron output and

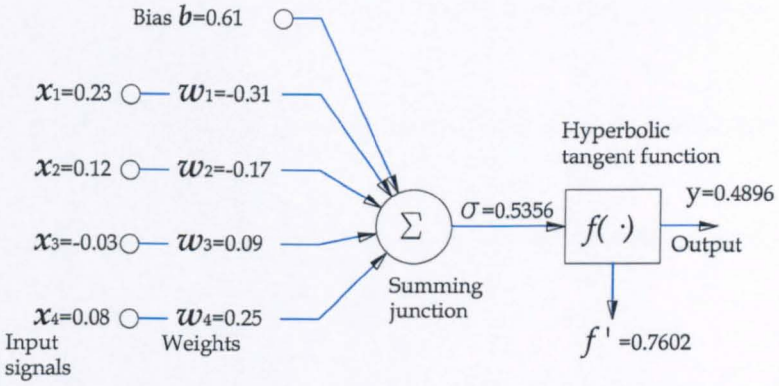


Figure 3.6: Example values for the different elements of an ANN.

the first derivative which is used in the training process by the backpropagation algorithm described later in this chapter.

3.8 Artificial neurons in a network

Used alone, an artificial neuron is not widely applicable. Its power and utility emerges when it is linked to other neurons to form a *network*. There is a countless number of ways in which they can be connected as well as the amount of neurons and the algorithms employed to train them. A large number of them have been developed by different researchers: multilayer feedforward networks, Hopfield networks, radial basis function networks, etc. Lekkas et al. (2001); Maier and Dandy (1998); Haykin (1999); Fine (1999) and many others introduce, explain and give examples of their applications.

The first step in introducing the concept of a network related to artificial neurons is given in the following definition by Haykin (1999):

> *A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing*

experiential knowledge and making it available for use. It resembles the brain in two respects:

1. *Knowledge is acquired by the network from its environment through a learning process.*
2. *Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.*

Among the different networks developed to date, there are two kinds well suited to prediction and forecasting applications: feedforward and recurrent networks, Maier and Dandy (1998), Haykin (1999), Fine (1999). In the former, neurons send information only to neurons in the next layer, see Fig. (3.7); in the latter, neurons can send information to neurons in the next layer, the previous layer, the same layer and even to themselves, see Fig. (3.8).

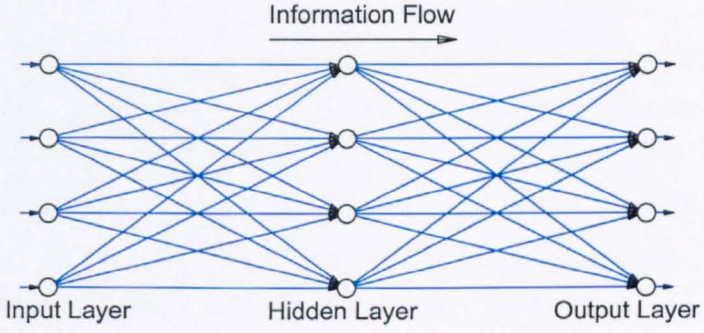


Figure 3.7: Fully connected feedforward network with one hidden layer.

Feedforward networks with no hidden layers (layers between the input and output layer) are considered, from a statistical point of view, generalised linear models. The presence of hidden layers, on the other hand, enables the network to extract higher-order statistics, Fine (1999). Feedforward networks have both the ability to model autoregressive components and, compared with recurrent networks, demonstrate better processing speed.

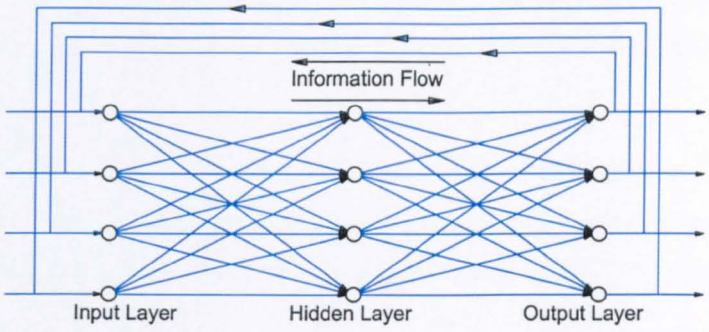


Figure 3.8: Partially connected recurrent network with one hidden layer.

Recurrent networks are characterised by the presence of at least one *feedback loop*. This feature enables this kind of network to do *temporal processing*, and become *dynamical systems*, although their cost is the instability associated with them.

In this thesis, the network that has been chosen to work with is the feedforward network. This option has been taken because, besides the points indicated above, the temporal processing required in simulating a hydrological model is carried out before the data is introduced into the neural network and, after testing both types of ANNs, there was not significant improvement to justify the greater computational power required by recurrent networks.

3.9 Data preprocessing

ANNs have a wide scope of applications in the hydrological area; therefore, the variables to be used may have different value ranges. Also they may have different statistical properties like covariance and mean and, consequently, in order to be used in an ANN environment, the input and output signals must be preprocessed.

Three measures commonly applied to data sets before they are introduced into an ANN model (Haykin, 1999) are depicted in Fig. 3.9 and described here:

- Standardisation. When ANNs work with different variables, these usually span different ranges. In order to ensure that all variables receive equal attention by the ANN, they must meet the same amplitude as well as the minimum and maximum values.
- Scaling. Due to the use of activation functions, signals must be scaled down to the value range covered by them. In addition, in order to avoid saturation of the activation function, scaling must be smaller than both of its extreme limits. For example, in this work the activation function ranges from -1 to 1, and the data sets have been scaled down to [-0.9 to 0.9].
- Normalisation to zero mean or a very small value compared to its standard deviation. This preprocessing avoids the shift of the first layer's synaptic weights to only increase or decrease values during the training process, Haykin (1999).

The other two preprocessing options proposed in this work for improving the ANN model are described in the following two sections. They are *Average Rainfall* and *Rainfall Filtering.*

3.10 Average rainfall

The impact of accumulated rainfall on a catchment must always be considered in any model intended to simulate the effect of rainfall on it. This element is the most important factor affecting a catchment behaviour, as described by Ahmad and Simonovic (2001). Physical models rely on parameters like vegetation, soil type,

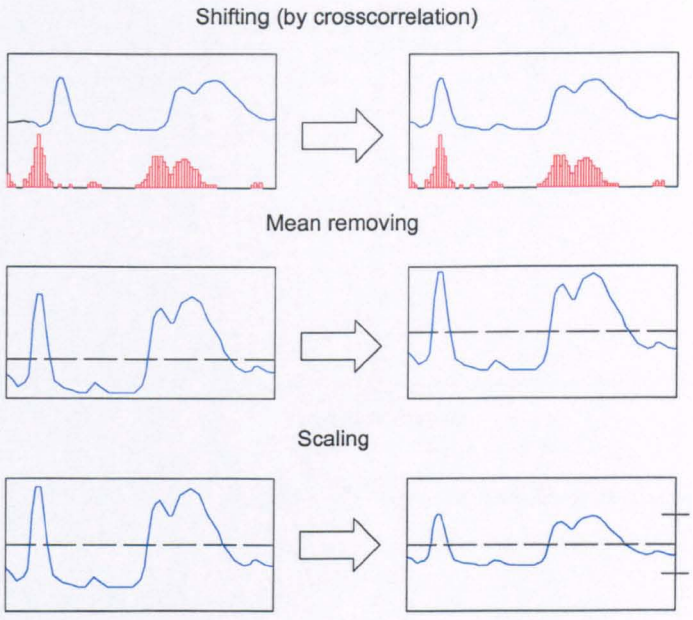


Figure 3.9: Preprocessing of the input signals.

seepage, soil moisture, etc. to determine the manner in which previous rainfall affects the actual runoff. ANN models, on the other hand, based on data series, use several techniques to represent the previous rainfall. Sajikumar and Thandaveswara (1999) enumerate three methods: window of input series, recurrent networks and the temporal back-propagation neural network. One of the methods used in this work considers the average of a number of previous rainfall values. This quantity is a moving average value with fixed length and has been fed to the ANN model as one of its inputs. An example of the effect of average rainfall on the model's efficiency is shown in Fig. 3.10 [3]. This plot presents the ANN model simulating the Brue catchment which is described later in this chapter.

[3]The definition and formula for *efficiency* are presented later in this chapter. Its ideal value is 1

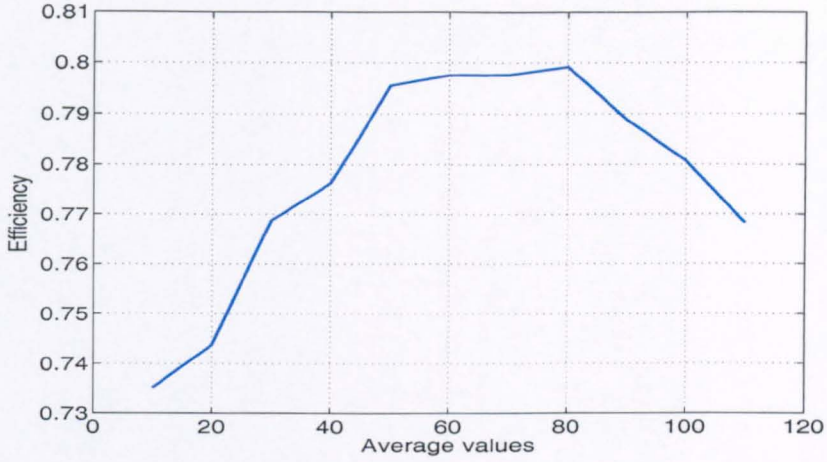


Figure 3.10: Effect of average rainfall on the model's performance.

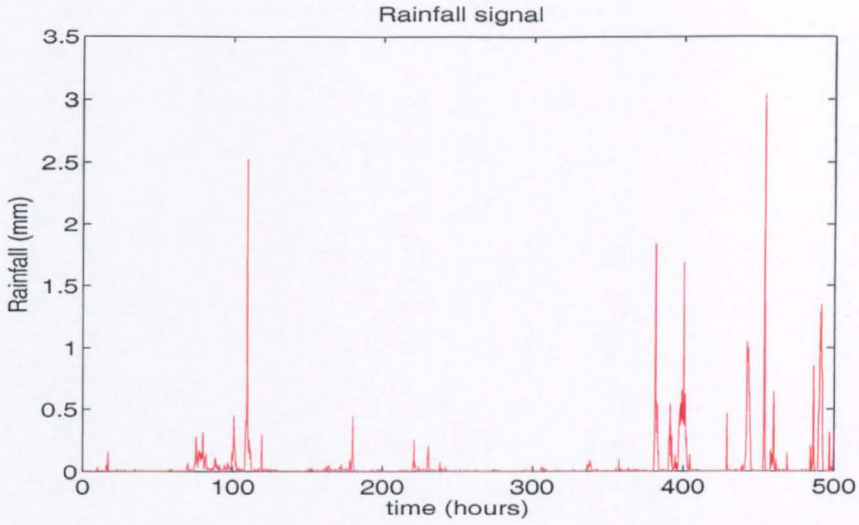
3.11 Rainfall filtering

Besides its chaotic behaviour, a rainfall data series is characterised by its high frequency and extreme variability. By contrast, runoff data are marked by their smoother profile. Examples of rainfall and runoff series were taken from HYREX[4]. The data set covered the period 01-Jan-1995 00:15 to 21-Jan-1995 20:15. The series are plotted in Fig. 3.11 for a) a hyetograph and b) a hydrograph.

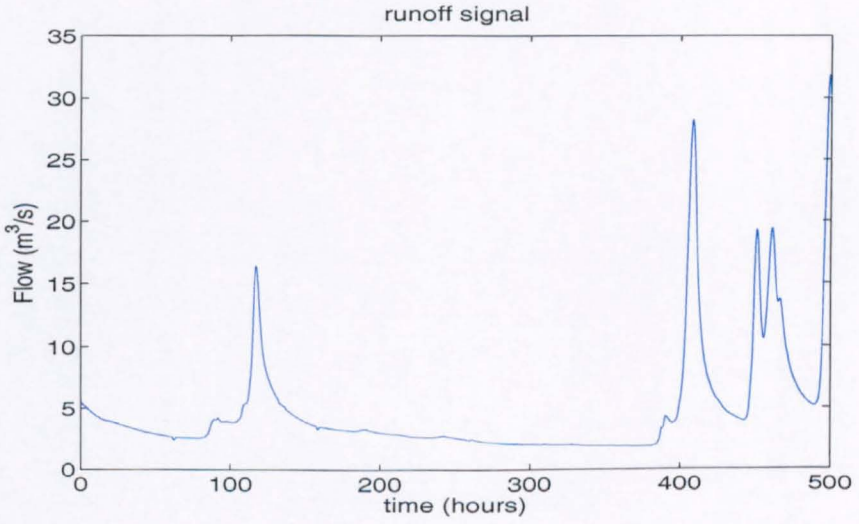
A useful technique employed to analyse the frequency properties of rainfall and runoff data is the transformation to the *frequency domain*. In this domain the frequency components of the series plotted in Fig. 3.11 are presented as normalised frequency responses in Fig. 3.12 a) for rainfall and b) for runoff. As can be seen in these graphs, the rainfall plot contains high frequency components not present in the runoff plot.

Though ANNs are considered lowpass filters on their own (Haykin, 1996, 1999), a lowpass filter is designed and applied to smooth this data series to ease the work

[4]This experiment is described later in this chapter



(a)

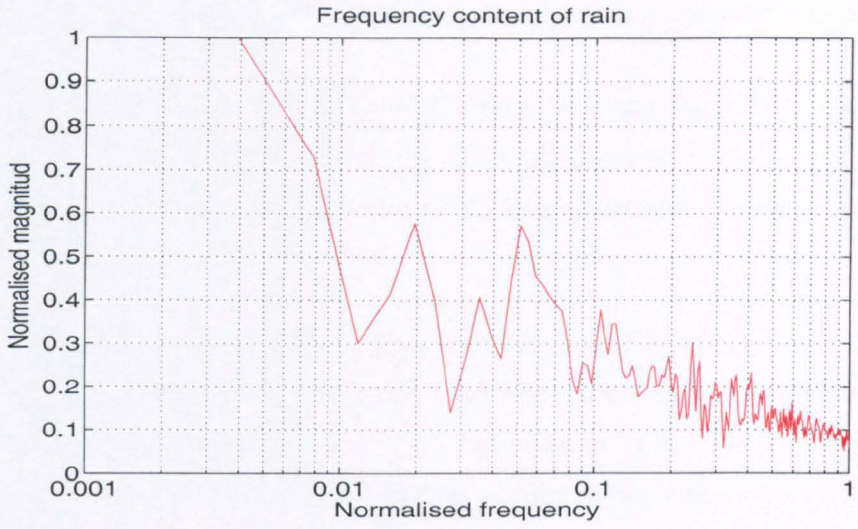


(b)

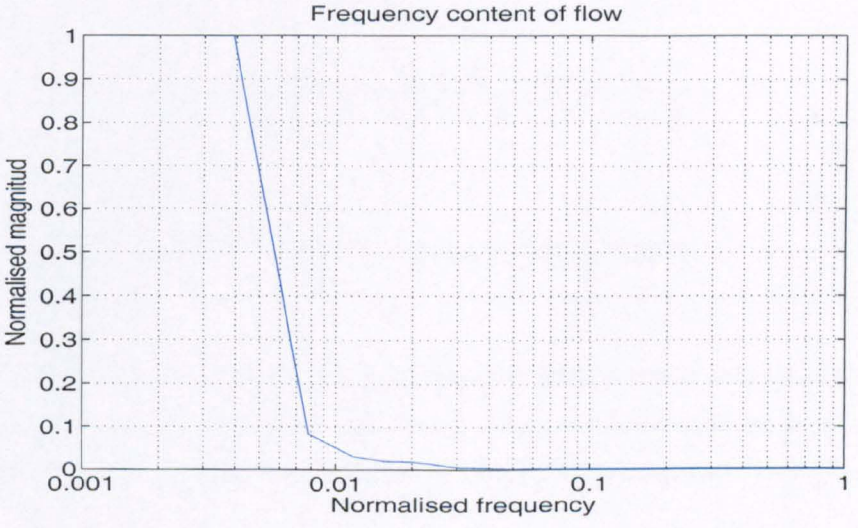
Figure 3.11: Example of (a) a hyetograph and (b) a hydrograph.

performed by the ANN model and improve its overall performance.

The first step in designing a digital filter is choosing between *Finite Impulse Response* (FIR) and *Infinite Impulse Response* (IIR) filters. In this thesis an IIR



(a)



(b)

Figure 3.12: Example of (a) rainfall and (b) runoff in the frequency domain.

filter has been chosen due to the fact that it requires fewer coefficients (Proakis and Manolakis, 1992; Mitra, 2001). The number of coefficients is related to the phase delay in such a way that more coefficients generate a bigger phase delay. The model has been intended to be applied to real time flood forecasting, hence, the objective is

to reduce the phase delay, working with the fewer number of coefficients as possible.

The second step is to define the cutoff frequency. From Fig. 3.12 it is possible to select a specific value, but because it is not possible to infer how the ANN model would perform with the resulting filter, several values have been selected in order to compare their resulting performance.

Having specified the cutoff frequency the next step is designing a digital filter in the discrete time-domain that fits the frequency response. A least-squares fit has been applied to approximate the IIR digital filters to the specified frequency response. The two general transfer functions applied in this process were Eqs. 3.12 and 3.13

$$H(z) = \frac{y(z)}{x(z)} = \frac{b_1 z^{-1} + b_2 z^{-2} + ... + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}} \tag{3.12}$$

$$H(z) = \frac{y(z)}{x(z)} = \frac{\frac{1}{n}}{1 - \frac{1}{n} z^{-1} - \frac{1}{n} z^{-2} ... \frac{1}{n} z^{-(n-1)}} \tag{3.13}$$

Fig. 3.13 presents an ideal frequency response using a cutoff frequency equal to 0.05 in the normalised frequency and the frequency response obtained by the filter using the transfer function 3.12 with the coefficient values expressed in Eq. 3.14.

$$H(z) = \frac{y(z)}{x(z)} = \frac{\left[2.879 - 13.41z^{-1} + 28.84z^{-2} - 36.76z^{-3} - 30.24z^{-4}\right}{1 - 7.403z^{-1} + 24.1z^{-2} - 45.05z^{-3} + 52.9z^{-4}} \cdots \frac{-16.41z^{-5} + 5.775z^{-6} - 1.218z^{-7} + 12.66z^{-8}\right] x10^{-5}}{-39.93z^{-5} + 18.93z^{-6} - 5.149z^{-7} + 0.6154z^{-8}} \tag{3.14}$$

The previous process has been done applying the cutoff frequencies 0.05, 0.2 and 0.4 in the normalised frequency. The rainfall and runoff data series used in these runs for testing the model are those displayed in Fig. 3.11. The ANN model

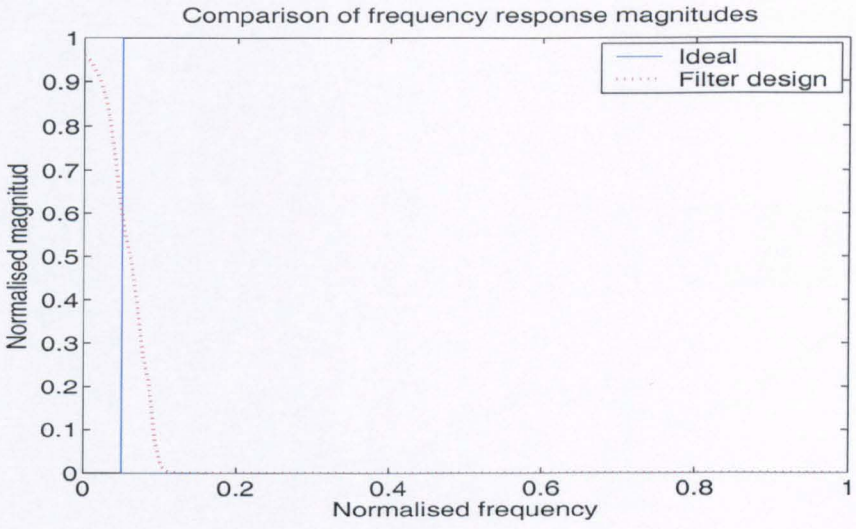


Figure 3.13: Ideal frequency response and filter designed.

had a 5-average efficiency value, without applying any filter to rainfall data, of $R^2 = -0.46029$. All the tests have been done using a 7 hr lead-time which is the approximate natural response time for the studied catchment. Fig. 3.14 shows the efficiency results, generated by the ANN model, run 5 times for every n-coefficient, hence they are 5-value averages. The 0.05 cutoff frequency graph produced negative average-efficiency values which remained higher than without a filter. The graphs for 0.2 and 0.4 cutoff frequencies, although mostly positive, were still not good enough and so several other parameters were tested, along with the other general transfer function (Eq. 3.13) to find an acceptable solution.

From the several tests performed, the filter with the best performance was the one described by the transfer function 3.15:

$$H(z) = \frac{y(z)}{x(z)} = \frac{\frac{1}{18}}{1 - \frac{1}{18}z^{-1} - \frac{1}{18}z^{-2} \ldots \frac{1}{18}z^{-(17)}} \tag{3.15}$$

The frequency response for this filter is displayed in Fig. 3.15.

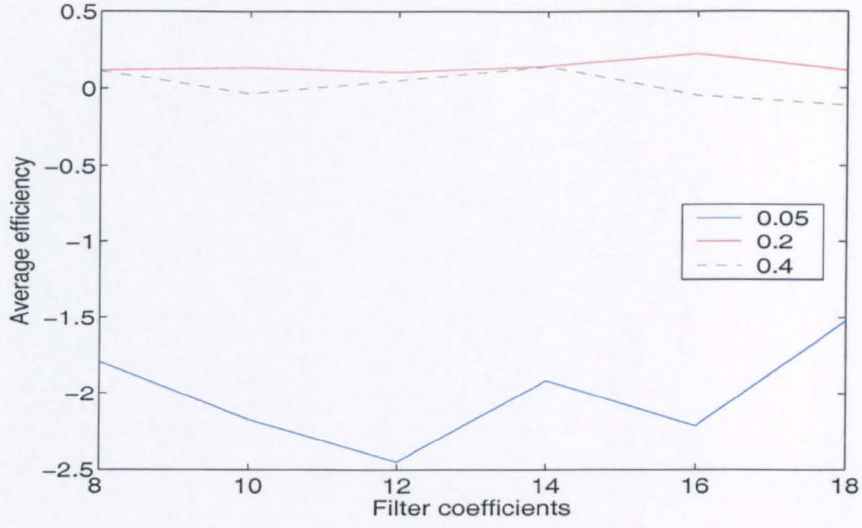


Figure 3.14: ANN model results for several cutoff frequencies and filter coefficients.

Fig. 3.16 presents the efficiency results for this filter and others using the same general transfer function. The 5-average efficiency was 0.809 for the 18-coefficient filter. The improvement, compared to previous tests is noteworthy and justified the use of this preprocessing tool in the ANN model.

Because all the analyses were made using a 7 hr lead-time, in order to visualise the impact of the phase-delay by the signal preprocessing, this 18-coefficient filter was tested using several lead-times. Fig. 3.17 presents the results and also demonstrates how the phase delay affected the model's performance.

The catchment's natural time response, from rain to flow measure is around 8.25 hours, corroborated by a crosscorrelation between rainfall and runoff series as it is plotted in Fig. 3.18 (see C for a description of this mathematical method). The natural time response would be the model's peak performance but, as it is shown in the Fig. 3.17, the peak performance was shifted by minus two hours, which is what could be expected by applying an 18-coefficient filter (Mitra, 2001).

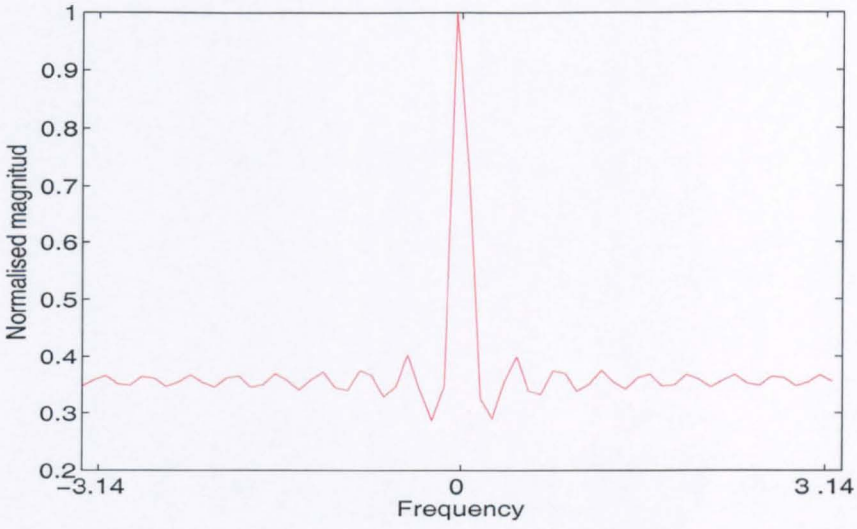


Figure 3.15: Frequency response by filter defined by Eq. 3.13.

3.12 Learning process

The property that mainly characterises a neural network is its ability to *learn* from examples (i.e., from its environment). The neural network adjusts its synaptic weights through an interactive process of getting inputs and checking outputs in accordance with some prescribed criteria.

The learning process is divided in two categories: *supervised learning* and *unsupervised learning*:

- **Supervised Learning** The network is feed with both, input and output (desired) data. The network is trained with what is expected to emit as output.
- **Unsupervised Learning** (or Self-Organisation) In this method the network is supplied only with input data. The network detect certain properties from the data set and learns to present them at the output. Unsupervised Learning usually performs the same task as an auto-associative network, compressing

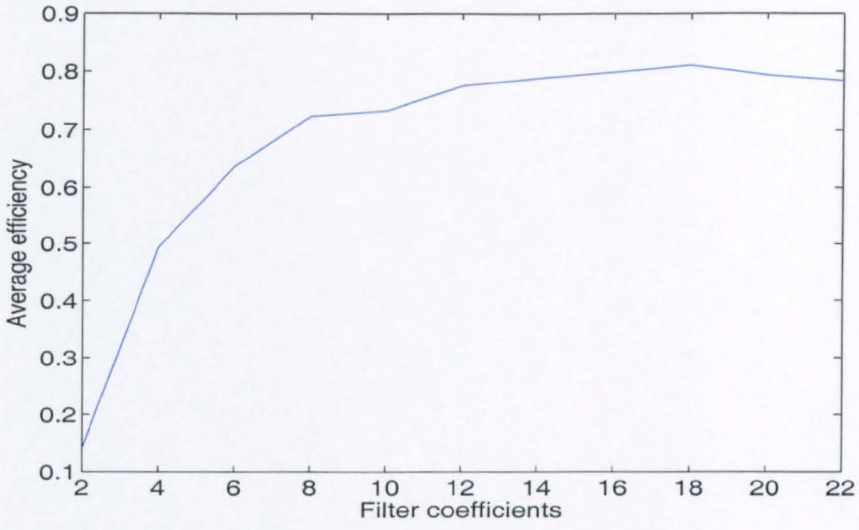


Figure 3.16: Efficiency results for different coefficients in the filter defined by Eq. 3.13.

the information from the inputs (Deco and Obradovic, 1996). Ripley (1996) points out its use for data visualisation.

In the learning process, the sample data is usually split into two parts: training and validation sets, Maier and Dandy (2000), but there is a trend that has been gaining acceptance to subdivide them in three parts: training, validation, and test sets.

Each one of these subsets has specific purpose and the distinctions are crucial.

Ripley (1996) defines each one in the following way:

- Training set: A data set used for learning, that is, to adjust the weights and bias of the network.
- Validation set: A data set used to determine the moment to stop the training process. When the error function of this data set reaches its minimum the process must stop to avoid overtraining.

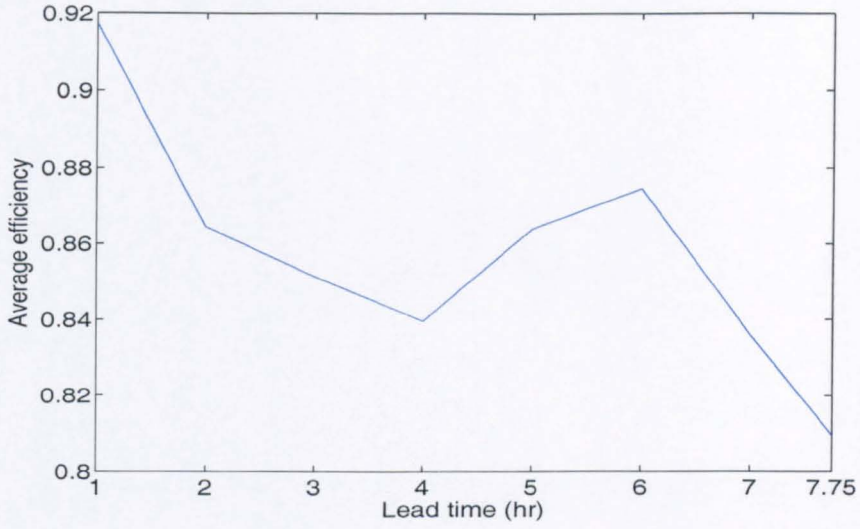


Figure 3.17: Efficiency results for different lead-times.

- Test set: A set of examples used only to assess the performance (generalisation) of a fully-trained network.

Several algorithms have been proposed to update the weights. The weights are adjusted to reduce the error function. One of these algorithms, commonly used on time series like rainfall-runoff relationship, is the *backpropagation algorithm*. Before presenting this algorithm the following subsection introduces the concept of *generalisation*.

3.12.1 ANN generalisation

After a learning process, when input x_i, and output (target) y_i data, where $x \in X$ and $y \in Y$ are used to train the ANN with, it is expected that the ANN can estimate values o_i where $o \in Y$, and is different from those the ANN was trained with.

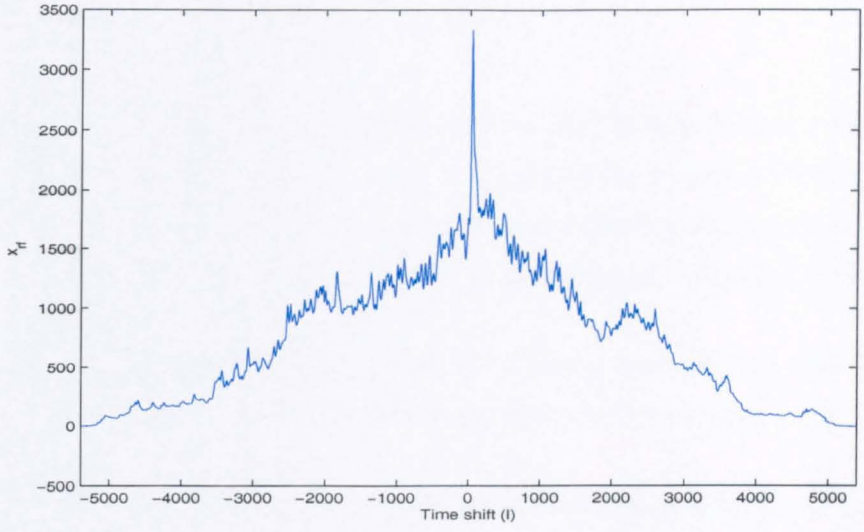


Figure 3.18: Crosscorrelation of rainfall and runoff series.

The performance level of an ANN on the universe of possible data (X, Y) (not just on the training data) is called its *generalisation ability* (Cheng and Titterington, 1994).

For a network to generalise effectively, it must be trained on a suitably diverse and large data set. There are several proposals for defining the amount of data needed to train an ANN. It is generally related to the number of connection weights, and hence the ratio between the number of training samples and the number of connection weights is the factor most broadly employed. This ratio varies according to the authors given values of 1, 2, 10 or even 30 (Maier and Dandy, 1998; Rogers and Dowla, 1994). In this thesis the data sets employed to train the ANNs were at least 10 times the number of connection weights.

3.12.2 Backpropagation algorithm

As described by its name, this training method goes backwards in its task of updating the weights. The 'standard backpropagation algorithm'is actually the *generalised delta rule.* A thorough and clear presentation of the convergence properties of the standard backpropagation method is found in Bertsekas and Tsitsiklis (1996).

The following analysis has been done for a neural network composed of two layers, although the explanation may be generalised to three or more layers.

Employing the neural network structure considered before, N input, output and desired vectors, $\mathbf{x}$(i), $\mathbf{y}$(i), and $\mathbf{o}$(i) respectively, form the following matrices:

$$\begin{aligned} X &= [\mathbf{x}(1) \ \ldots \ \mathbf{x}(i) \ \ldots \ \mathbf{x}(N)] \quad \text{q x N matrix} \\ Y &= [\mathbf{y}(1) \ \ldots \ \mathbf{y}(i) \ \ldots \ \mathbf{y}(N)] \quad \text{r x N matrix} \\ O &= [\mathbf{o}(1) \ \ldots \ \mathbf{o}(i) \ \ldots \ \mathbf{o}(N)] \quad \text{r x N matrix} \end{aligned} \tag{3.16}$$

These matrices are the complete series of vectors used for the whole training process. Here, q and r are the number of samples to be used in each iteration and N is the number of iterations.

If the activation function $f(\sigma)$ is represented with the Greek letter phi ϕ, the complete set of the output and hidden vectors can be calculated as:

$$Y = \phi_y(W^y \cdot H) \ ; \ H = \phi_h(W^h \cdot X) \tag{3.17}$$

The algorithm starts with the error calculation of the output y generated by the network and the desired output o, Eq. 3.18.

$$\varepsilon(i) = \mathbf{o}(i) - \mathbf{y}(i) \tag{3.18}$$

where each element is an $r \times 1$ vector. This output error serves to find a set of weight matrices, W^h, W^y, such that the error is as small as possible.

The performance index used here is the **mean-squared error**, $J(W^h, W^y)$, defined as:

$$J(W^h, W^y) = \frac{1}{rN} \sum_{i=1}^{N} E(W^h, W^y, i) \tag{3.19}$$

where $E(W^h, W^y, i)$ is defined as:

$$E(W^h, W^y, i) = \frac{1}{2} \sum_{j=1}^{r} \varepsilon_j^2(i) \tag{3.20}$$

The weight update, according to the **steepest descent learning algorithm**, should be proportional to the gradient of the *mean-squared-error function* of Eq. (3.19).

$$\nabla J(W^h, W^y) = \frac{1}{rN} \sum_{i=1}^{N} \nabla E(W^h, W^y, i) \tag{3.21}$$

here, $\nabla E(W^h, W^y, i)$ is defined as

$$\nabla E(W^h, W^y, i) = \left[\frac{\partial E}{\partial w_{11}^h} \cdots \frac{\partial E}{\partial w_{kj}^h} \cdots \frac{\partial E}{\partial w_{Kq}^h} \cdots \frac{\partial E}{\partial w_{11}^y} \cdots \frac{\partial E}{\partial w_{lk}^y} \cdots \frac{\partial E}{\partial w_{rK}^y} \right] \tag{3.22}$$

There are two methods for applying the steepest descent algorithm: the **pattern** and **batch** training algorithms. In the former the weight update is done after every iteration of samples is presented to the neural network. In the later the weight matrices are updated after the presentation of all the iterations. In this thesis the pattern procedure has been used, and so it is described in the following analysis.

For the last layer, the gradient component can be calculated as follows:

$$\begin{aligned}
\frac{\partial E(i)}{\partial w^y_{lk}} &= -\varepsilon_l \frac{\partial y_l}{\partial w^y_{lk}} \quad \text{where } E = \frac{1}{2}(... + \varepsilon_l^2 + ...) \text{ and } y_l = \phi(\sigma^o_l) \\
&= -\varepsilon_l \frac{\partial y_l}{\partial \sigma^o_l} \frac{\partial \sigma^o_l}{\partial w^y_{lk}} \quad \sigma^o_k = W^y_l \cdot \mathbf{h} = ... + w^y_{lk} h_k + ... \\
&= -\varepsilon_l \cdot \phi'_l \cdot h_k \quad \phi'_l = \frac{\partial y_l}{\partial \sigma^o_l} \\
&= -\delta^y_l \cdot h_k \quad \delta^y_l = \varepsilon_l \cdot \phi'_l
\end{aligned} \tag{3.23}$$

For the hidden layer the gradient component is slightly more complicated:

$$\begin{aligned}
\frac{\partial E(i)}{\partial w^h_{kj}} &= -\sum_{l=1}^{r} \varepsilon \frac{\partial y_l}{\partial w^h_{kj}} \quad \text{where} \quad y_l = \phi(\sigma^i_l) \\
&= -\sum_{l=1}^{r} \varepsilon_l \frac{\partial y_l}{\partial \sigma^i_l} \frac{\partial \sigma^i_l}{\partial w^h_{kj}} \quad \sigma^i_l = ... + w^y_{lk} h_k + ... \\
&= -\left(\sum_{l=1}^{r} \delta^y_l w^y_{lk} \right) \frac{\partial h_k}{\partial w^h_{kj}} \quad \delta^y_l = \varepsilon_l \cdot \phi'_l \quad h_k = \phi_h(W^h_{:kj} \cdot \mathbf{x}) \\
&= -\left(\sum_{l=1}^{r} \delta^y_l w^y_{lk} \right) \phi'_{h \cdot k} \cdot \mathbf{x}_j \quad \phi\prime_{h \cdot k} = \frac{\partial \phi_{h \cdot k}}{\partial w^h_{kj}} \\
&= -\delta^h_k \cdot \mathbf{x}_j \quad \delta^h_k = \left(\sum_{l=1}^{r} \delta^y_l w^y_{lk} \right) \phi'_{h \cdot k}
\end{aligned} \tag{3.24}$$

Finally, the weight and bias updates for the last and hidden layers are:

$$\begin{aligned} W^y(i+1) &= W^y(i) + \eta \cdot \delta^y(i) \cdot \mathbf{h}^T(i) \\ W^h(i+1) &= W^h(i) + \eta \cdot \delta^h(i) \cdot \mathbf{x}^T(i) \\ \beta^{y,h}(i+1) &= \beta^{y,h}(i) + \eta \cdot \delta^{y,h}(i) \end{aligned} \tag{3.25}$$

where η is the **learning rate** parameter. The algorithm seeks a trajectory in the weight space toward the best set of weights, hence, η provides an adjustment to the magnitude of the update. This adjustment is performed seeking to get an agreement between a big value of η, that accelerates the process but where the system may become unstable and a small value of η which generates a smoother trajectory but at a slower learning rate.

This process is repeated for every one of the $(\mathbf{x}_i, \mathbf{y}_i)$ pairs constituting the training set. This training set is repeated n times (*epochs*), until the error (cost) function falls below a predefined level, or by using data series to validate the learning process which is the option selected in this thesis.

3.12.3 Accelerating the learning process

The algorithm described above is usually slow when implemented in practical situations. There are several modifications applied in order to accelerate the learning process. Three of these modifications imply a change and an addition to the basic steepest descend algorithm:

- **Variable Learning Rate** A fixed learning rate which is too small makes the learning process very slow. A learning rate which is too large makes it oscillate about the desired optima. A variable learning rate is the logical approach to avoid both extremes. The strategy applied in this thesis was to increase the

learning rate if the error function, after each epoch, has decreased in relation to its previous value (less than 0.95 times); and to decrease it if the total error function exceeds the old error by more than a pre-specified ratio (more than 1.05 times) (Cichocki and Unbehauen, 1993). Fig. 3.19 presents the behaviour of the variable learning rate η for a fix 100-epoch learning process.

- **Momentum** The inclusion of momentum, though represents a minor modification to the algorithm, stabilises the section of the learning process when it has sign oscillations. It also prevents the process from terminating in a shallow local minimum on the error surface.

- **Resilient Backpropagation** The widely used sigmoidal and hyperbolic tangent functions have the characteristic that, close to their extremes, the slope tends to zero though the input could be very large. It implies a small change in the weights even though the weights and biases are far from their optimal values.

  This algorithm eliminates the use of the magnitudes of the partial derivatives. Instead it uses the sign of the derivative. The size of the weight change is determined by a separate update value. This update value is increased by a fixed factor whenever the derivative of the activation function has the same sign for two successive iterations. Likewise, the update value is decreased by a fixed factor whenever the derivative of the activation function changes sign from the previous iteration.

There are other methods implying the use of standard numerical optimisation techniques. They are focused in the numerical optimisation of the error function (when the error function includes other elements such as penalties for weight decay, it is usually called *objective function*). Some of these methods are *conjugate-gradient*, *quasi-Newton*, and *Levenberg-Marquardt*, (Bertsekas and Tsitsiklis, 1996; Cichocki and Unbehauen, 1993). The choice is based on the characteristics of the problem being dealt with, for example, the total number of weights.

3.13 Performance criteria

A fundamental element of the ANN performance criteria is the evaluation of the error. The output signal is a function of the connection weights, the activation function and the network geometry. This output is compared with the desire output signal. It implies the use of one of several formulae. Two of the most commonly applied are **RMSE** (Root Mean Square Error) and R^2 efficiency (Nash and Sutcliffe, 1970)

$$RMSE = \sqrt{\frac{\sum_{k=1}^{K}(Q_k - \hat{Q}_k)^2}{K}} \tag{3.26}$$

$$R^2 = \frac{F_o - F}{F_o} \tag{3.27}$$

in which

$$F_o = \sum_{k=1}^{K}[Q_k - \bar{Q}]^2 \tag{3.28}$$

$$F = \sum_{k=1}^{K}[Q_k - \hat{Q}]^2 \tag{3.29}$$

where k is a dummy time variable, K is the number of elements in the considered period, Q_k and $\hat{Q}_k$ are the observed and the computed runoffs at the kth time interval respectively, and $\bar{Q}$ is the mean value of the runoff for the same period.

This review of the ANN fundamentals, though not exhaustive, explains the core of the ANN model employed in simulating the hydrological processes analysed in this thesis.

The following section describes the model used to simulate rainfall-runoff relationships before presenting the results obtained with such a model.

3.14 Model description

As every catchment has its particular characteristics, like size, altitude variations, type of soil and vegetation, rainfall rate, etc., the ANN model and any model intended to simulate their rainfall-runoff relationships must be an adaptable system in order to match them as accurately as possible. Therefore, the ANN model is not a fixed system but rather a system able to modify its architecture and parameters according to the catchment it is intended to simulate.

The selection of the optimal architecture and values of several parameters was done by a genetic algorithm which is described in the next chapter.

The genetic algorithm searched for the best options of the variables involved between limits specified previously. These limits were defined according to recommendations found in literature, analysis of convenience and tests performed with the model using values beyond such limits to evaluate its performance.

The parameters specified by the genetic algorithm were number of layers, neurons in each layer, number of average inputs and number of average rainfall values for those average inputs. Table 3.1 presents the range of values the genetic algorithm made the selection from.

The other parameters in the model had a fixed value indicated in Table 3.2 and only η was adjusted during the training process.

The model was implemented in C++ and Matlab. C++ code, besides generating faster programs, allowed tracking of the behaviour of several parameters like the learning rate. The flow diagram of the algorithm developed in this work is described in Appendix A.

The application of a variable learning rate, as indicated previously, accelerates the learning process by applying a large value at the beginning and diminishing it

Table 3.1: Range of parameter values the GA selected from to build the ANN Model used in rainfall-runoff relationships

Parameter	Range
Number of layers	3-5
Neurons in each layer	4-26 in increments of 2 (1 in the last layer)
Average Input	0-2
Average rainfall values	20-400 in increments of 10 for 15-minute time steps 2-8 in increments of 1 for 6-hour time steps

Table 3.2: Other parameter values used by the ANN Model

Parameter	Range	Tested
Number of initialisations	22	1-100
η learning rate	0.2-0.001 (variable)	0.8-0.00001
Momentum α	0.1	nil, 0.1, 0.2
Activation function	hyperbolic tangent	sigmoidal, linear

later on when a detailed searching of optimal values is required . Fig. 3.19 shows the behaviour of the learning rate along three different learning processes. The limits for η were set between 0.2 and 0.001.

Before presenting the results, an analysis of different time steps is presented in the following section.

3.15 Time-step analysis

Rainfall and runoff data from the Brue catchment were generated every 15 minutes but in order to decide what time-step was most suitable to maximise the ANN model's performance several time-steps were tested. The Brue catchment is a rel-

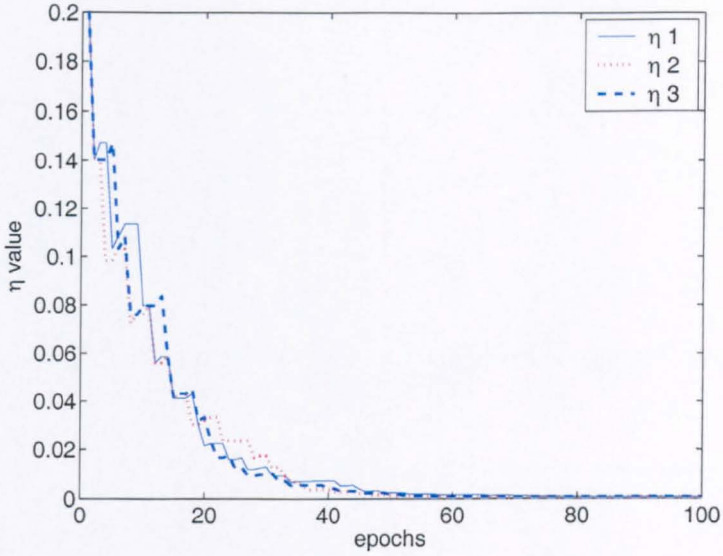
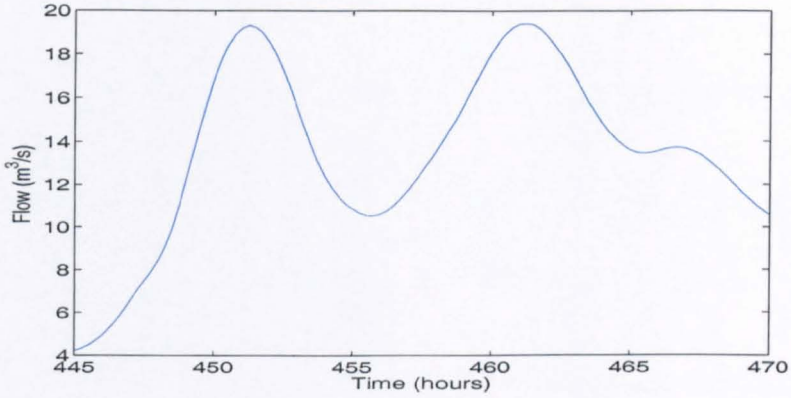


Figure 3.19: Variable learning rate behaviour during a training process.

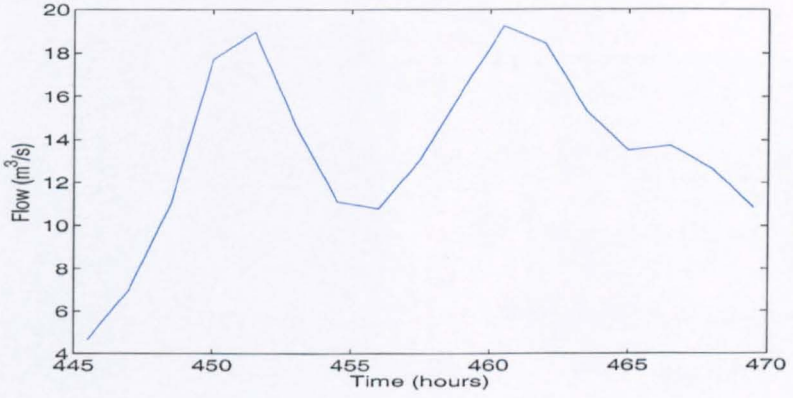
atively small catchment with a natural time response of around 8.25 hours, hence the time-steps tested ranged from 15-minute to 90-minute with a 15-minute separation among them. Longer time-steps were not considered because the longer the time-step the bigger the variations in the flow that would be lost. As an example see Fig. 3.20, which displays the same hydrograph but (a) is for a 15-minute step and (b) is for a 90-minute step.

The periods and number of values for each time-step are presented in table 3.3.

The variations in lead time are owed to the adjustment to the lower closest value in every time-step. The flow estimations are instantaneous values and hence, flow values are the same for the different time-step series at the same times. The rainfall estimations are cumulative values, therefore the values vary for every series at identical times. The use of a filter on the rainfall data requires the visualisation of their components in the frequency domain. For this reason the spectrum of rainfall data for every time-step was computed and is displayed in Fig. 3.21.



(a)



(b)

Figure 3.20: Hydrographs representing the same period but at different time step: (a) 15-minute time step, (b) 90-minute time step.

The plots show that all the series have components over the whole spectrum and hence, the filtering is still required. To identify which filter order generated the best performance for each time-step, several orders were used in the simulations. The plot presenting the model's performance using all the time-steps for several filter orders is shown in Fig. 3.22.

In this plot the 90-minute time-step has the highest efficiency with an eight-order filter, but the lead-time for this time-step, as indicated in Table 3.3 is only 6 hours.

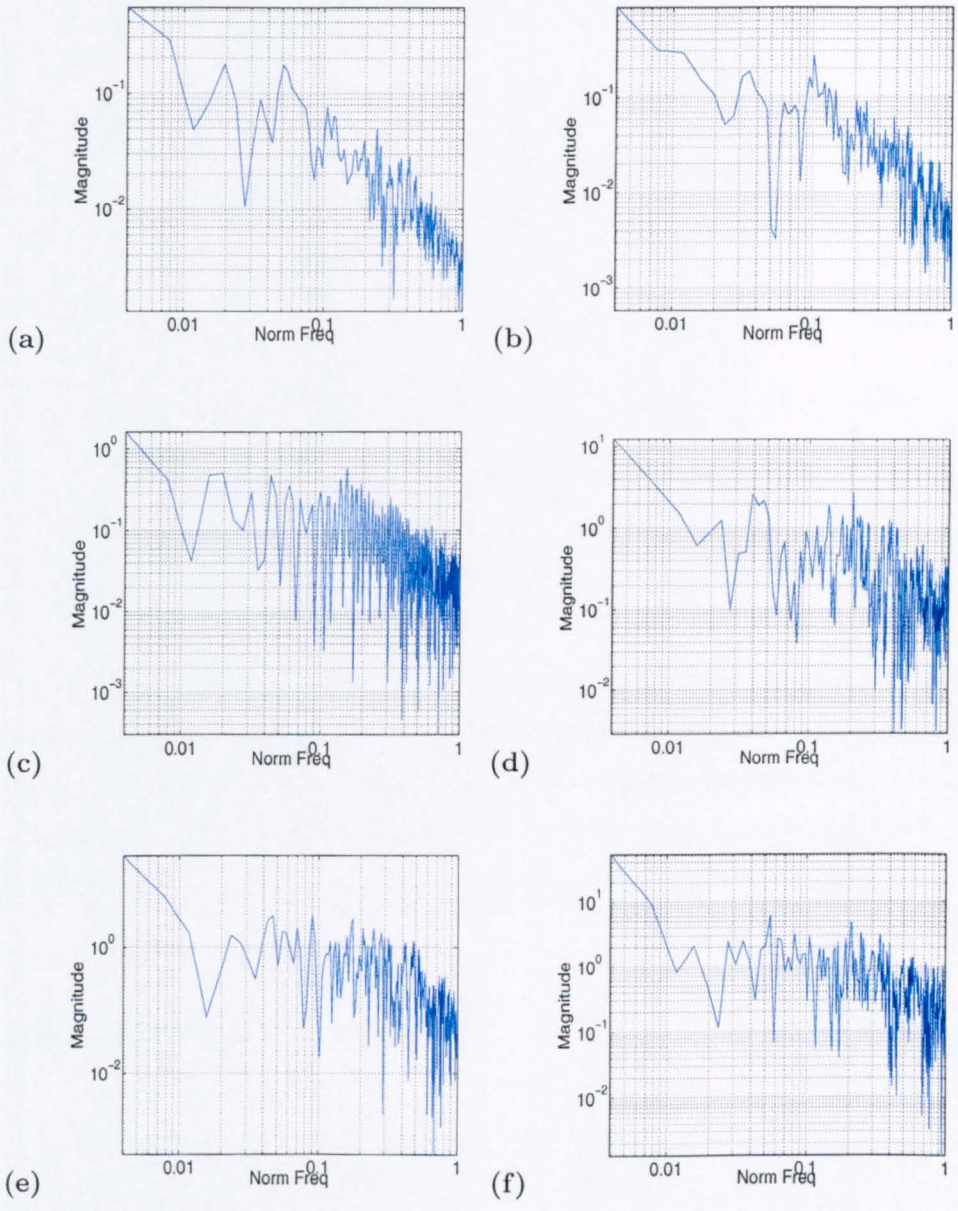


Figure 3.21: Frequency representation of hyetographs for different time-steps (in minutes): (a) 15, (b) 30, (c) 45, (d) 60, (e) 75, and (f) 90.

Table 3.3: Data values for the time-steps applied to study the model's performance

Periods (year 1995)	15-min	30-min	45-min	60-min	75-min	90-min
Training 01-Jan 00:15 05-Feb 11:00	3404	1702	1134	851	680	567
Validation 05-Feb 11:15 15-Feb 21:00	1000	500	334	250	200	167
Testing 15-Feb 21:15 26-Feb 07:00	1000	500	334	250	200	167
Average rainfall values	80	40	27	20	16	14
Lead time (hours)	7	7	6.75	7	6.25	6

Running the model with a 15-minute time-step, using a 6-hour lead time, makes its performance greater, hence the 15-minute time-step has been chosen for simulating the Brue catchment. Better performance means a shorter lead-time, and a larger lead-time generates a lower efficiency. This trade-off would be a decision made by the modeller and, hence, both results will be included in the following section.

3.16 Results

The ANN model was tested using data from two catchments: the Brue catchment in the UK and the Bird Creek catchment in the U.S.A. The data from the former has a 15-minute time-step, the adequate value for testing the model which is intended to perform on a real-time basis generating estimations every 15 minutes. The later, with its 6-hr time-step, was used to study the model's performance on a long term scale.

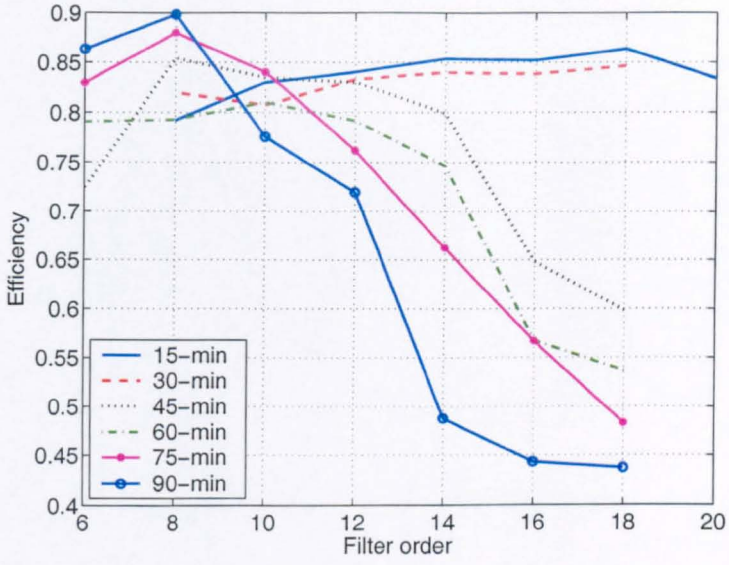


Figure 3.22: Model's efficiency for different time-steps against filter order.

A Transfer Function (TF) model has been used to compare the ANN model's performance. This TF model has been chosen because it uses rainfall and runoff data only. The description of this model is given in the Appendix B.

3.16.1 Brue catchment

With the ANN model described in the previous section, the first test was performed using rainfall and runoff data from the Brue Catchment in Somerset. This is a small catchment draining an area of 135 km^2 and rising from 20 m to 260 m AOD, see Fig. 3.23. The rainfall and runoff data from this catchment are part of the **HY**drological **R**adar **EX**periment (HYREX) (Moore et al., 2000). The radar data were obtained by a dense network of 49 raingauges installed in the catchment having at least one recording gauge in every 2 km square. The river flow was measured at Lovington.

The natural time response of the rainfall-runoff relationship for this catchment

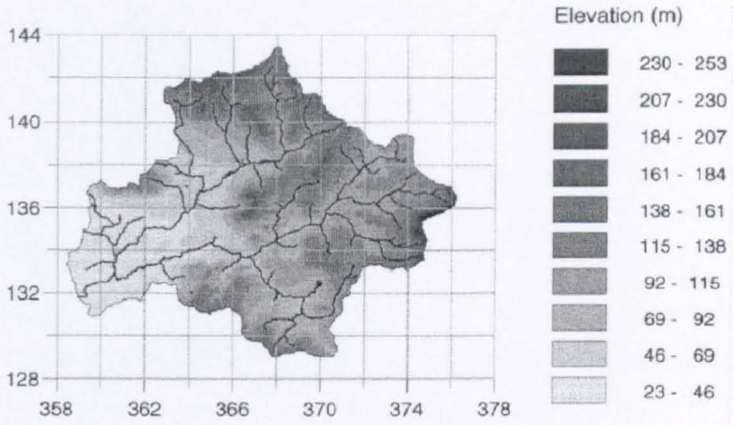


Figure 3.23: Relief over the Brue catchment.

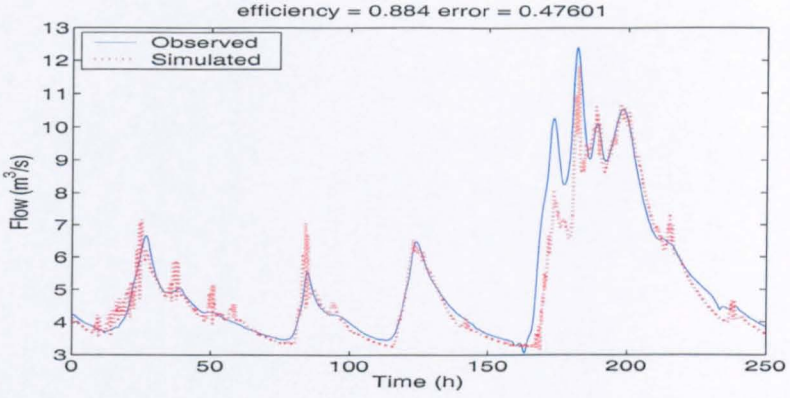
is around 8:00-8:45 hours and the lead times used for running the model were 6 and 7 hours due to the effect of the filter phase-delay.

The architecture and parameter values selected by the GA for this catchment are presented in Table 3.4.

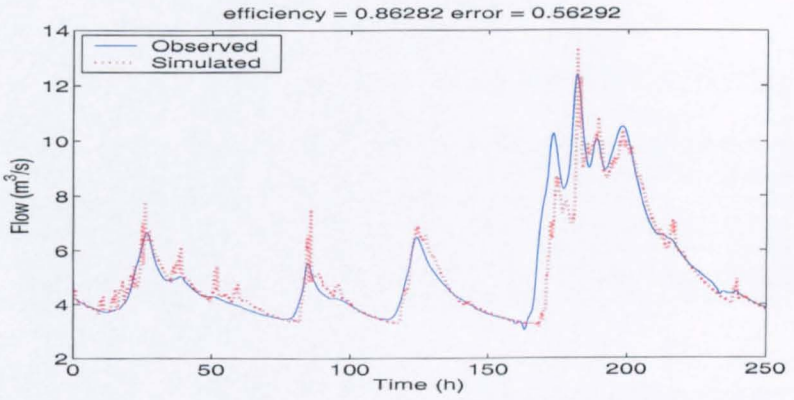
Table 3.4: Parameter values obtained by the GA to build the ANN Model used for the Brue catchment data

Parameter	Value
Number of layers	4
Neurons in each layer	10 - 8 - 8 - 1
Average Input	1
Average rainfall values	80

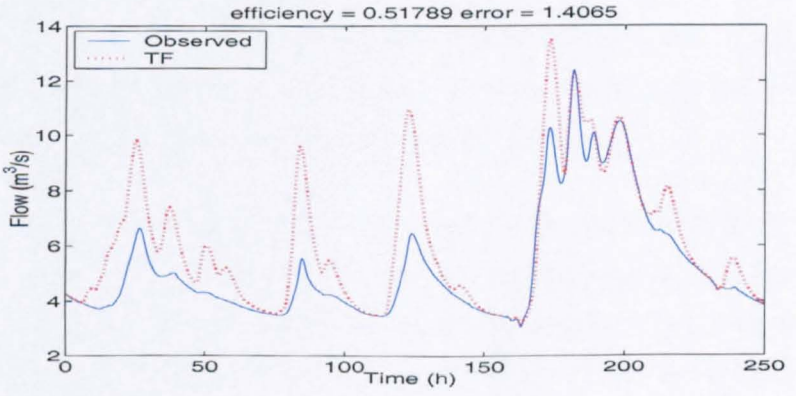
The first run performed by the model covered the time periods presented in Table 3.5. These months are considered wet months due to the rainfall intensity registered in this catchment. The result of this run is displayed in Fig. 3.24. The efficiencies for the test period were $R^2 = 0.863$ and $R^2 = 0.884$ for the 7-h and 6-h lead-times respectively, and $R^2 = 0.52$ for the transfer function model.



(a)



(b)



(c)

Figure 3.24: Observed and simulated flow using data from the Brue catchment: (a) 6-h lead-time, (b) 7-h lead-time, (c) transfer function model.

Table 3.5: Training, validation and test data sets for year 1995

Set	Time (hr)	Start time	End Time
Training	875	01-Jan 0:00	05-Feb 11:00
Validation	250	05-Feb 11:15	15-Feb 21:00
Test	250	15-Feb 21:15	26-Feb 07:00

A second data set from the same Brue catchment is presented in Table 3.6. This run was performed during the wet months, but in a different year: 1995 for the ANN training and 1996 for the validation and test periods.

Table 3.6: Training, validation and test data sets for the second test on the Brue catchment

Set	Time (hr)	Start time	End Time
Training	375	01-Jan-1995 00:15	16-Jan-1995 15:00
Validation	250	01-Jan-1996 00:15	08-Jan-1996 19:30
Test	250	08-Jan-1996 19:45	08-Jan-1996 15:00

Before describing the results generated by the ANN model, the rainfall and runoff series are plotted in Fig. 3.25. Although the month is the same (January), the rainfall series from 1995 shows a smaller rainfall rate but with a higher single event. The rainfall series from 1996 presents a higher rainfall rate but with values less than a half of the highest value in the previous year.

The results, shown in Fig. 3.26, correspond to the second half of the series in year 1996. Due to the significant difference in rainfall rate and intensity between the time periods employed, the ANN model simulated higher flow peaks than those observed; nevertheless, the efficiency for this period was $R^2 = 0.909$ for the 6-h lead-time, $R^2 = 0.891$ for the 7-h lead-time, and $R^2 = 0.841$ for the transfer function model.

A third example with data from the Brue catchment used time periods from

nonsuccessive years. The training period was from 1996 and the validation and test periods were from 1998 as shown in Table 3.7.

Table 3.7: Training, validation and test data sets for the third test on the Brue catchment

Set	Time (hr)	Start time	End Time
Training	375	06-Feb-96 11:15	22-Feb-96 02:00
Validation	337.5	25-Mar-98 08:00	08-Apr-98 09:15
Test	337.5	08-Apr-98 09:30	22-Apr-98 10:45

The results from this third run is displayed in Fig. 3.27. The highest discrepancy between simulated and observed flow was $1.2m^3/s$ and the efficiency was $R^2 = 0.888$ for the 6-hr lead-time, $1.2m^3/s$ and $R^2 = 0.863$ for the 7-hr lead-time, and $5.8m^3/s$ and $R^2 = -0.572$ for the transfer function model.

A fourth example with data from the Brue catchment used time periods from successive years. The training period was from 1996 and the validation and test periods were from 1997 as shown in Table 3.8.

Table 3.8: Training, validation and test data sets for the fourth test on the Brue catchment

Set	Time (hr)	Start time	End Time
Training	575	15-Nov-96 19:15	09-Dec-96 18:00
Validation	475	11-Nov-97 14:00	01-Dec-97 08:45
Test	475	01-Dec-97 09:00	21-Dec-97 03:45

The resulting graphs from this fourth run is displayed in Fig. 3.27. The efficiencies were $R^2 = 0.927$ for the 6-hr lead-time, $R^2 = 0.902$ for the 7-hr lead-time, and $R^2 = 0.841$ for the transfer function model.

The selected periods were separated in wet and dry periods in order to obtain a better performance by the ANN model. Next is an example of using periods covering dry and wet seasons which affect the ANN model's performance.

The training period covers the first half of 1995, the validation period covers just over a month and the test period covers five months, as it is described in Table 3.9.

Table 3.9: Training, validation and test data sets for fifth test on the Brue catchment

Set	Time (hr)	Start time	End Time
Training	4500	01-Jan-95 00:15	07-Jul-95 12:00
Validation	825	01-Jan-96 00:15	04-Feb-96 09:00
Test	3675	04-Feb-96 09:15	06-Jul-96 12:00

The result, for a 7-hr lead-time is shown in Fig. 3.29. The decrease in efficiency is clear and establishes the complexity of a catchment system.

The ANN model was tested on other periods from the Brue catchment but they were used for comparative purposes in Chapter 5.

3.16.2 Bird Creek catchment

The Bird Creek catchment is located in the state of Oklahoma close to the northern state border with Kansas in the U.S.A., see Fig. 3.30. This catchment covers an area of 2344 km^2 and is relatively low lying with altitudes ranging from 175m up to 390m AOD (WMO, 1992). The catchment borders and the stream network are shown in Fig. 3.31.

The rainfall and runoff data formed part of a real-time hydrological model intercomparison exercise conducted in Vancouver, Canada in 1987 and reported in WMO (1992). These data series have a time step of 6 hours. The filter order was 4 and, the training, validation and test periods are indicated in Table 3.10.

The efficiency value R^2 for the test period was 0.75826 and the plot of the hydrograph for the same period is displayed in Fig. 3.32.

Table 3.10: Training, validation and test data sets for Bird Creek catchment

Set	Time (hr)	Start time	End Time
Training	5856	17-Nov-1972 12:00	19-Jul-1973 06:00
Validation	1566	14-Sep-1973 18:00	18-Nov-1973 18:00
Test	1560	19-Nov-1973 00:00	22-Jan-1973 18:00

Although the time span for running the ANN model was large, the 6-hr time step meant that there were not so many values to run the model with. This inconvenience meant a lack of choice of data to train the ANN model with more efficiency, according to seasonal periods. Nevertheless, as can be seen in Fig. 3.27, the highest discrepancy between the observed and simulated flows was after the receding limb of the highest peak in this period.

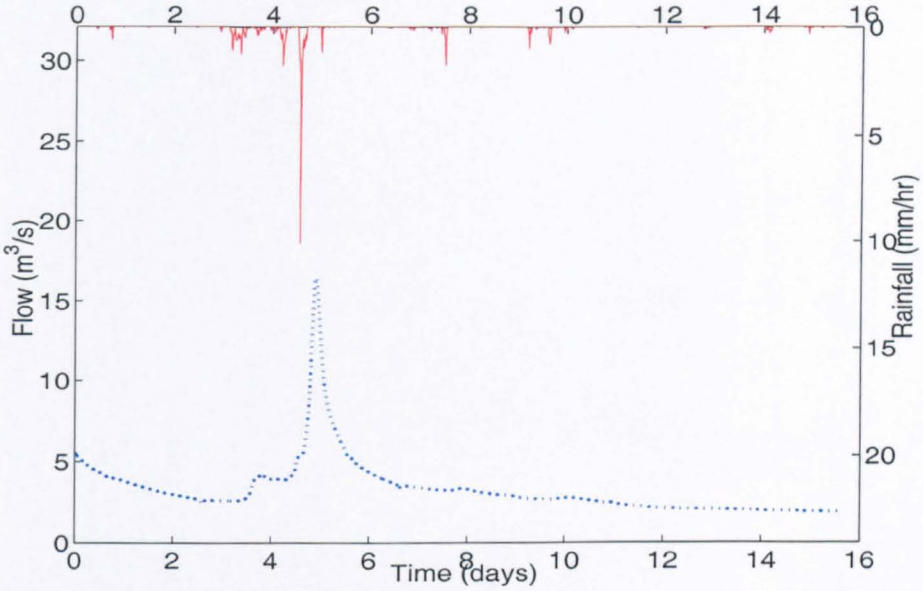
3.17 Discussion

Working with ANNs requires a need to be aware of their potential and limitations. They are highly adaptable to different situations but need to be adjusted to the specific task at hand. A review of several ANN models applied to rainfall-runoff relationships is presented by Maier and Dandy (2000). It shows multiple applicable options for ANNs based on the features of each problem, such as type, time step and quantity of data, catchment characteristics and authors preference. In this chapter the main aim has been to provide the basic structure for a real-time forecasting model, updating every 15 minutes, using rainfall and runoff data, and forecasting at the largest lead-time, that is, the catchment natural response time between where rainfall is recorded, and the flow, generated by that rain, is measured.

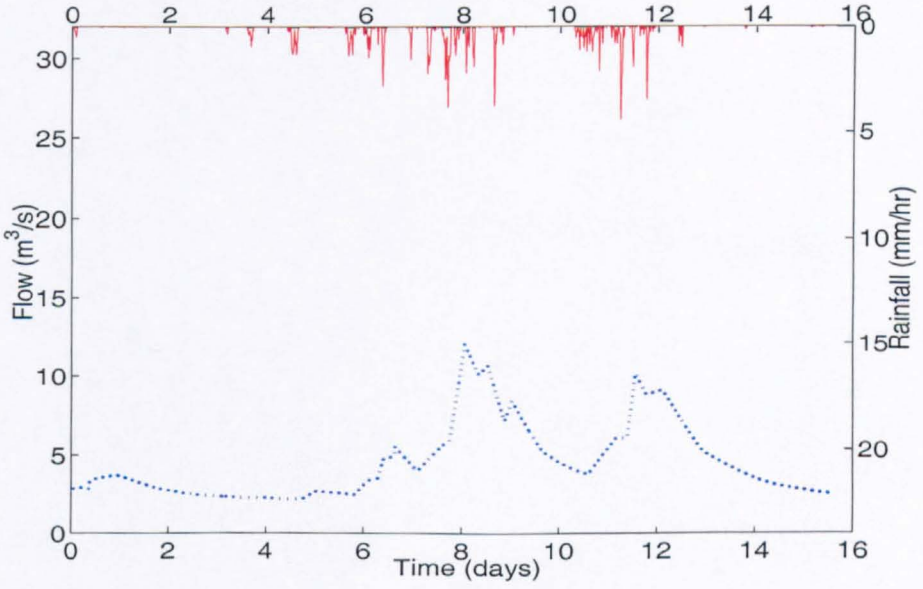
The application of preprocessing techniques has been a decisive factor in improving the ANN model's performance. Signal filtering has been the most important technique of all. Even considering the fact that applying a filter to a data

series involves a phase delay, the improvement in accuracy exceeds the reduction in lead-time forecast. If the lead-time is left equal to the catchment natural response, though the accuracy diminishes compared to the point shifted by the phase delay effect (see Fig. 3.17), it still has a higher value than the non-filter option (as seen in Fig. 3.16). The point to note is that, like other parameters in an ANN model, the number of filter coefficients must be defined for every particular situation.

The use of average rainfall as an estimation of a catchment wetness however, does not take into account other hydrological elements like evapotranspiration and infiltration, but helps to improve the model's performance. In fact, because seasonal changes in temperature, vegetation, evapotranspiration, etc., have a substantial effect on a catchment's runoff response (Ward and Robinson, 1990), it is recommended, whenever possible, to use rainfall and runoff data from the previous year to train a neural network model for every season. This will facilitate the learning process and at least reflect any fundamental changes in the input-output process experienced to date.

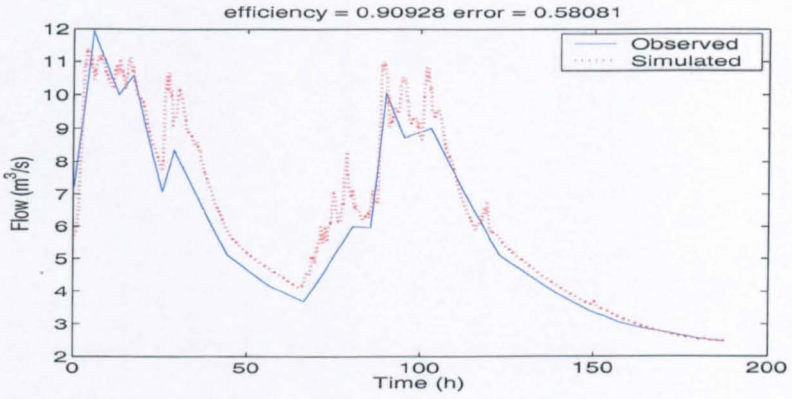


(a)

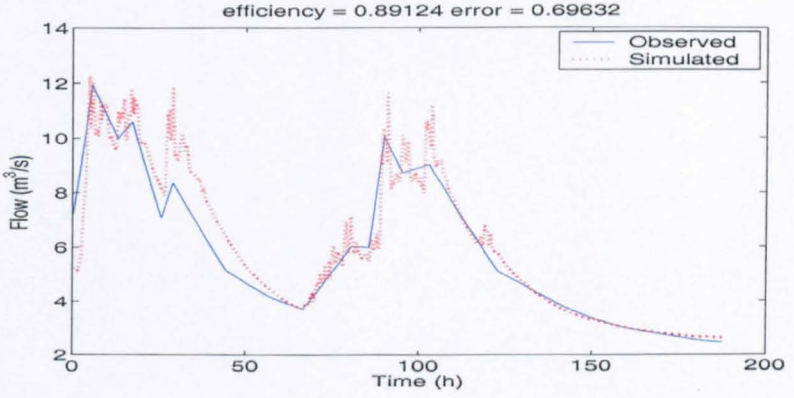


(b)

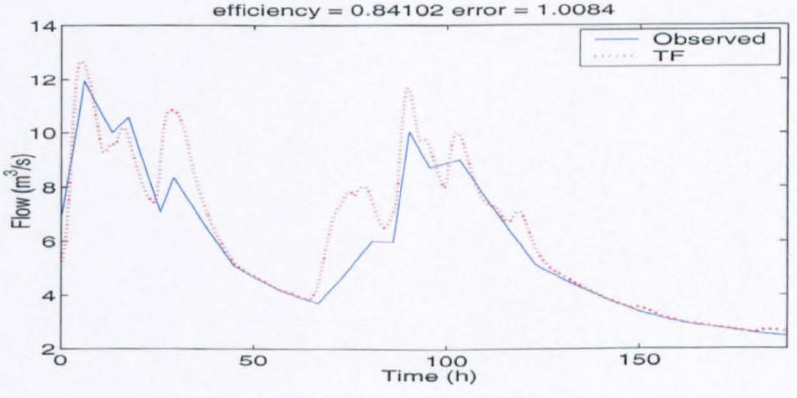
Figure 3.25: Hyetographs and hydrographs from January (a)1995 and (b)1996.



(a)

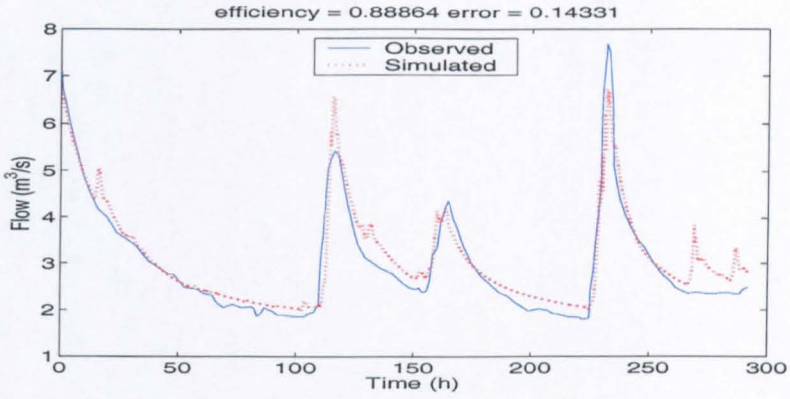


(b)

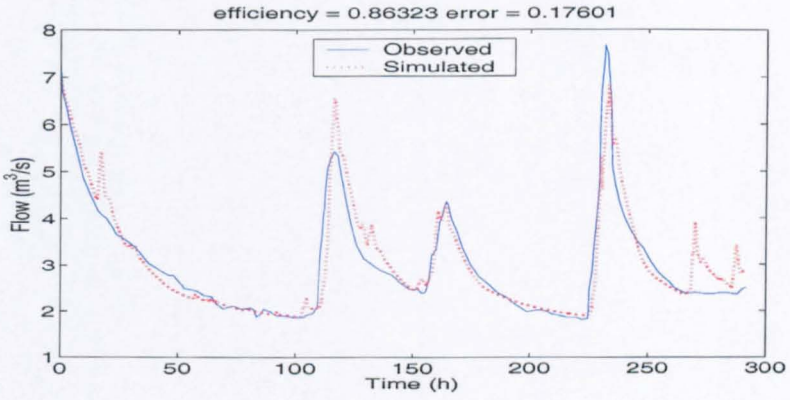


(c)

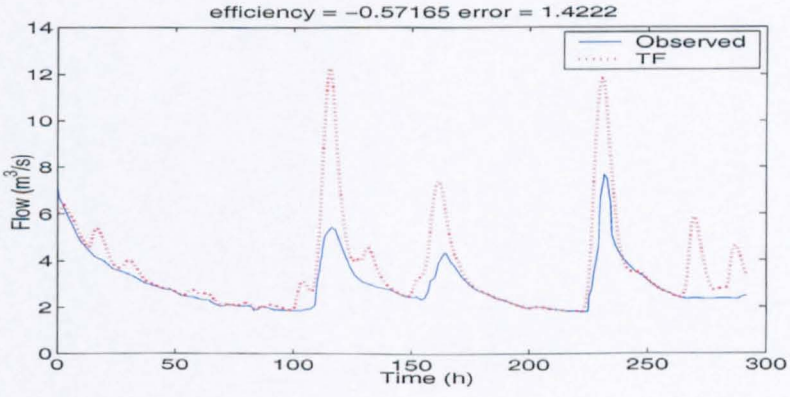
Figure 3.26: Second set of observed and simulated flow using data from the Brue catchment: (a) 6-h lead-time, (b) 7-h lead-time, (c) transfer function model.



(a)

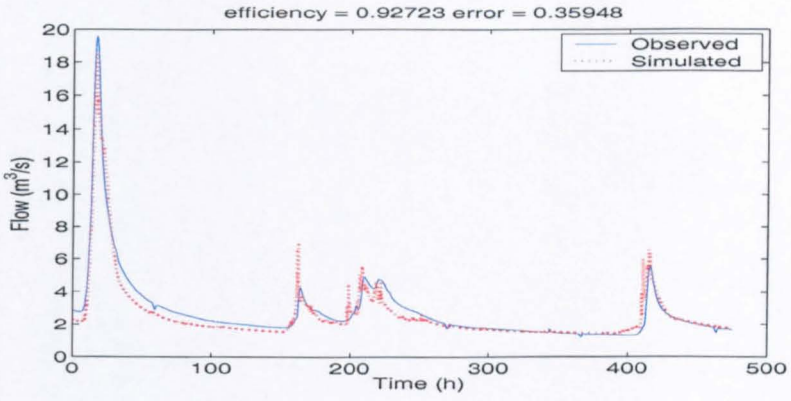


(b)

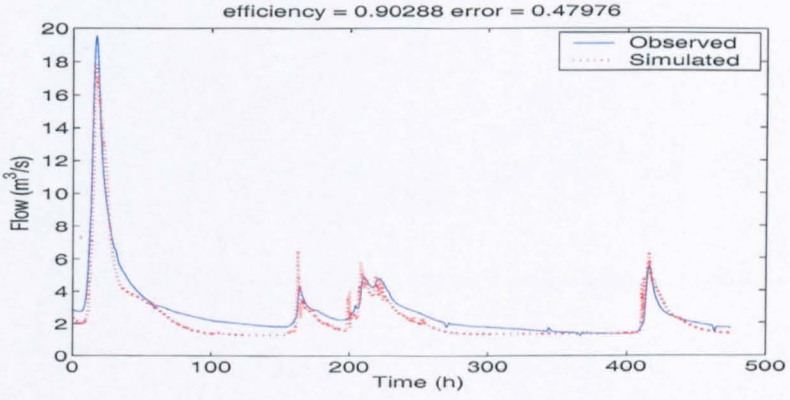


(c)

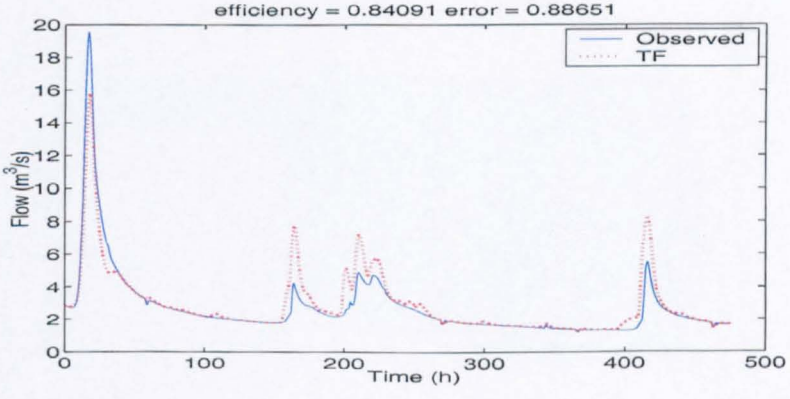
Figure 3.27: Third set of observed and simulated flow using data from the Brue catchment: (a) 6-h lead-time, (b) 7-h lead-time, (c) transfer function model.



(a)



(b)



(c)

Figure 3.28: Fourth set of observed and simulated flow using data from the Brue catchment: (a) 6-h lead-time, (b) 7-h lead-time, (c) transfer function model.

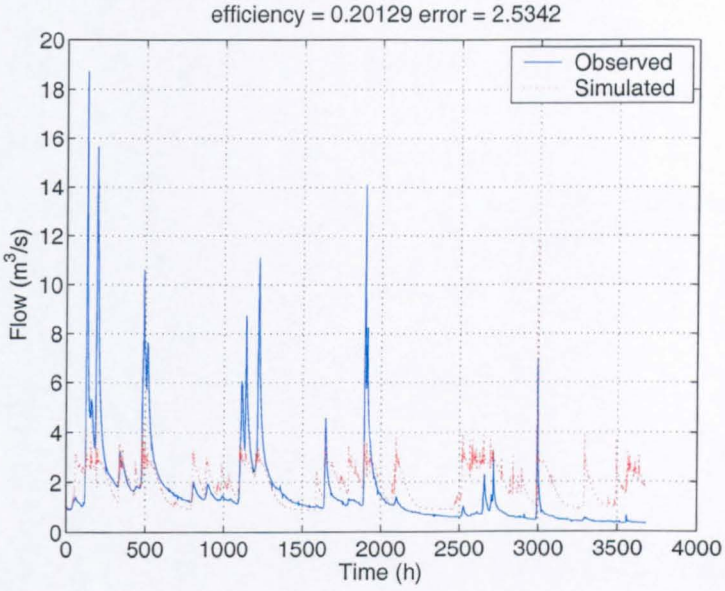


Figure 3.29: ANN model results for a half-year training period.

Figure 3.30: Location of the Bird Creek catchment (in blue) in Oklahoma state, U.S.A.

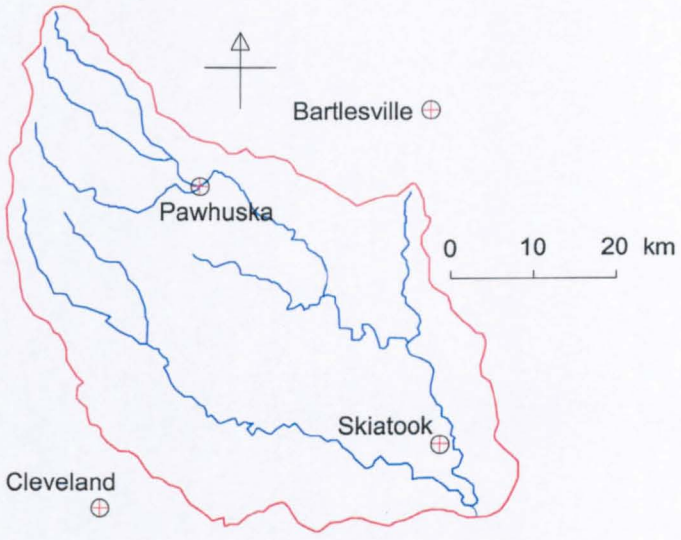


Figure 3.31: The Bird Creek catchment in Oklahoma state, U.S.A.

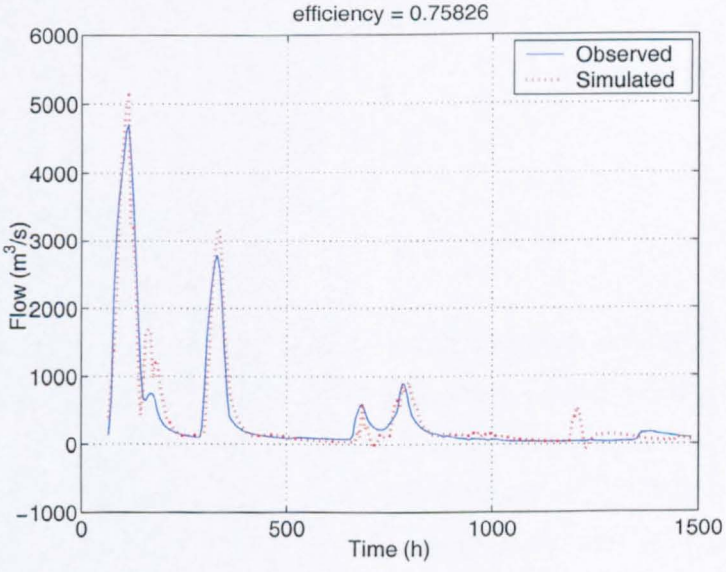


Figure 3.32: Observed and simulated flow data for the Bird Creek catchment.

Chapter 4

Improvements of the ANN model by a Genetic Algorithm

To err is human. To really foul up, use a computer

quoted by D. E. Goldberg

4.1 Introduction

Natural evolution processes have inspired the development of the numerical optimisation algorithms known as **Genetic Algorithms** (GAs). They can be used to search for a set of model parameters so that a predefined performance index is maximised or minimised. They were first clearly described by Holland (1975), and are based on both natural selection and natural genetics. GAs are algorithms that work through generations of individuals (possible solutions) expose to three operators, crossover, mutation and selection. One of their main characteristics is their robustness and, during the last two decades they have been successfully applied to

optimisation problems such as parameter estimation, adaptive control, and optimal control problems.

This chapter consists of two sections. The first section presents a brief introduction to GAs whilst the second section focuses on the application of the GAs on searching the best set of parameter values for the ANN model. Previous works have used GAs to optimise ANNs finding their number of layers and number of neurons in each layer. In this thesis two new parameters, number of average-rainfall layers, and number of average data, are introduced to improve the ANN model.

4.2 Basic concepts

GAs are complicated and powerful algorithms which search a function space for an optimal solution of a non-linear system.

Although there have been variants of genetic algorithms developed recently, with modifications and/or additions mainly applied to target specific applications, the basic structure remains the same. A simple typical GA starts with a set of potential solutions, selected from the entire population of possible solutions of a specific problem. Each possible solution is a string composed by a set of parameters (or genes) required by the problem, e.g.: x, y and z for a three-dimensional function, $F(x, y, z)$. The set of potential solutions (or individuals) passes a selection process based on fitness of individuals, that is, if the objective is to obtain the maximum value of F within specific limits, this function is evaluated using each individual's genes. Once all the individuals are evaluated they are arranged in an ascending or descending order. The best of them pass their genes on to the next generation via crossover. A mutation process is added affecting some of the individuals to widen diversity. The process then is repeated again

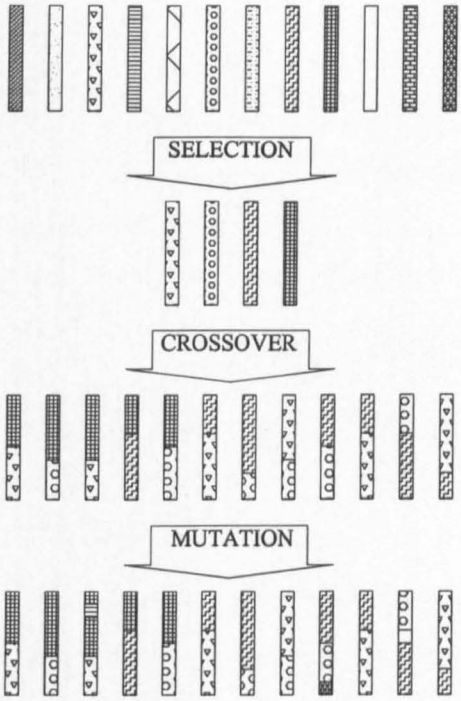


Figure 4.1: Basic Operators of a Genetic Algorithm.

GAs terminology is taken from biology, thus, following is a list of terms to refer to in the GAs context (Mitchell, 1996; Holland, 1975; Goldberg, 1989). The biological definition is accompanied by the computational definition in those terms where it applies:

- *DNA*: deoxyribonucleic acid. It is the molecule which contains the genetic information found in most organisms and is shaped like a double helix (a spiral).
- *Gene*: a length of DNA that encodes a particular protein. The computational counterpart is a digit position that can take on certain values, e.g. in a base-10 algorithm a gene can hold any number between 0 and 9.

- *Chromosomes*: a set of strings containing the instructions to build up and maintain an organism. This set is formed by multiple genes and is found within each cell of every living being. The GA term is simply a string of genes, e.g. in base-8 12345670 could be a chromosome of length 8 (10 in base-8).
- *Allele*: slightly different forms of the same gene.
- *Genotype*: the set of alleles possessed by an individual organism.
- *Phenotype*: the morphology, physiology, or behaviour of an organism. The GA counterpart is the parameter set to be used by the *fitness function* for evaluating each chromosome.
- *Locus*: the physical location of a gene in a chromosome.
- *mutation*: it is the action in which single nucleotides (elementary bits of DNA) are changed from parent to offspring. The GA term refers to flipping a digit in the chromosome after crossover.
- *Crossover*: the exchange of genes, during sexual reproduction, between two parents. In the GA context it is an operation which swaps genes between two chromosomes.
- *Schema*: a similarity template describing a subset of chromosomes with similar genes at certain locus.

GAs combine elements of direct and stochastic search and have been successfully applied to large-scale optimisation problems and real-valued parameter estimations within complex search spaces riddled with many local optima, see Fig. 4.2. GAs are different from more normal optimization and search procedures like hill-climbing, random search and iterated search, in four ways (Goldberg, 1989):

1. GAs work with a coding of the parameter set, not the parameters themselves.

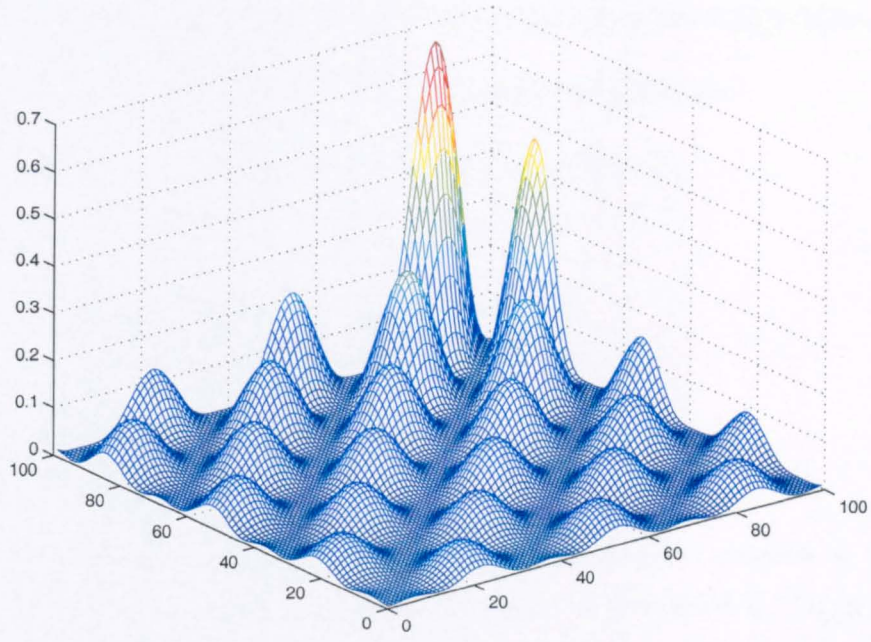


Figure 4.2: Solution-surface with many local optima.

2. GAs search from a population of points, not a single point.

3. GAs use payoff (performance index) information, not derivatives or other auxiliary knowledge.

4. GAs use probabilistic transition rules, not deterministic rules.

A sequence for a typical algorithm might be the following:

1. A genetic representation for potential solutions to the problem.

2. A starting population generated randomly where each individual is a potential solution.

3. A process to calculate the *fitness* or how bad or good every individual is within the population. This is using an evaluation function that plays the role of ranking the potential solutions.

4. A way to mix genes (crossover) from the better solutions to form new ones.

5. A mutation operator to seek diversity within the population.

6. A method to evaluate when to stop the steps (3) to (5).

4.3 Elements of the algorithm

4.3.1 Data structures

Considering the previous function F with three independent variables x, y and z: $F(x, y, z)$, a possible individual to be used in a GA could be (x_1, y_1, z_1) if $x_1, y_1, z_1 \in C[x, y, z]$. x_1, y_1 and z_1 can have different representations: binary, decimal, hexadecimal, etc. The representation proposed by Holland (1975) and commonly adopted in order to ease the application of operators is binary. For illustrative purpose, if $x = 13$, $y = 6$ and $z = 3$, four possible representations are the following:

- decimal: (13,6,3)
- binary: (1101,110,11)
- hexadecimal: (D,6,3)
- octal: (16,6,3)

4.3.2 Genetic operators

The core in any GA are the three operators mentioned previously: crossover, mutation and selection. These operators are inspired by evolution theory and have proved

to be effective in innumerable applications (Lakhmi and Martin, 1999). Following is a description of each operator:

CROSSOVER Reproduction is a process in which one or two organisms give birth to one or more individuals with similar characteristics.

MUTATION Several processes can cause natural mutation: radiation, chemical agents, errors during crossover, etc. In GAs, mutation is implemented using a probability function applied to every gene in the string.

SELECTION From Darwinian theories selection can be understood as the ability an individual has to survive and reproduce, on a particular environment over other individuals of the same species.

4.4 Theoretical foundation of genetic algorithms

This section presents an introduction to the basic theory in order to understand why genetic algorithms work. More information can be found in Mitchell (1996), Coley (1999), Holland (1975), Keane (1995), and Goldberg (1989), as well as many other papers, textbooks and internet sites.

First in this introduction is the notion of schema. By definition, a schema (Holland, 1975) is a similarity template describing a subset of strings with similarities at certain string positions. Without losing generality the following explanation is limited to the binary alphabet $\{0, 1\}$ and the use of the (*) or *don't care* symbol to indicate the possibility that, at the position where a (*) is found there could be a (1) or a (0).

A schema represents all strings which match it on all positions but on (*), e.g. for a 5-length string, a schema S is S=[0*1*1] and it matches the following four strings:

$\{00101, 00111, 01101, 01111\}$

It is clear that this schema matches 2^r strings, where r is the number of (*). For other types of string representation, the number of matched strings would be $(k)^r$ where k is the number of symbols in the alphabet used. If a binary 20-length schema, has 18 positions occupied by (*) it would match $(2)^{18}$=262,144. This example gives a brief idea of the magnitude of information a GA is expected to process, then, in order to manage the different types of schemata two important schema properties are introduced: order and defining length.

The order of a schema S, denoted by $o(S)$, is the number of specific (or fixed) bits (either 0 or 1) in the schema S. In the example above, the order of the schema [0*1*1] is 3 (denoted by o(0*1*1)=3). The defining length of a schema S (or $\delta(S)$) is the distance between the first and the last fixed string positions. It defines the compactness of information contained by the schema. For example, the schema 0*1*1 has a δ(S)=5-1=4.

Schemata and their properties are the basic means for studying the net effect of genetic operators on the whole population. There is, clearly, the need to evaluate these schemata in order to assign them some utility, or goodness value that is intended to be maximised. This is done by a function that evaluates the average fitness of all strings in the population matched by a determined schema S at a time t as follows:

$$f(S,t) = \frac{1}{k}\sum_{i=1}^{k} f(v_i) \tag{4.1}$$

where $(v_1, v_2, v_3, ..., v_k)$ are the k strings in the population matched by the schema S.

The first genetic operator to be analysed is reproduction. Through reproduction a string is copied according to its fitness, e.g. its probability p_i:

$$p_i = \frac{f(v_i)}{\sum_{j=1}^{k} f(v_j)} \tag{4.2}$$

where $f(v_i)$ is the fitness of the ith string and $\sum f$ is the fitness sum of all strings. If the number of all examples of a particular schema S within the actual population is $r = r(S,t)$ at time t, the number of S at time $t+1$ will be:

$$r(S,t+1) = \frac{r(S,t) \cdot k \cdot f(S,t)}{\sum_{j=1}^{k} f(v_j)} \tag{4.3}$$

if the average fitness of the entire population is written as $\overline{f} = \sum f_j/k$, then Eq. 4.3 can be written as follows:

$$r(S,t+1) = r(S,t)\frac{f(S,t)}{\overline{f}} \tag{4.4}$$

This equation expresses how a particular schema grows as the ratio of the average fitness of the schema to the average fitness of the population. This means that an above-average-fitness schema receives an increasing number of strings in the next generation.

For crossover the schema survival is based on the probability p_s that the cross site on the string falls outside the defining length $\delta(S)$, thus there will be a ratio between $\delta(S)$ and the whole string length $l-1$:

$$p_s = 1 - \frac{\delta(S)}{l-1} \tag{4.5}$$

If crossover is itself performed by random choice, say with probability p_c at a particular mating, the survival probability may be given by the expression

$$p_s \geq 1 - p_c \cdot \frac{\delta(S)}{l-1} \tag{4.6}$$

For the last operator, mutation, in order for a schema S to survive, all of the specified positions must themselves survive. Each allele survives with probability $(1-p_m)$, and as each of the mutations is statistically independent, a specific schema survives when each of the fixed positions within the schema survives, e.g. each probability multiplied $o(S)$ times:

$$p_t = (1 - p_m)^{o(S)} \tag{4.7}$$

Finally, considering the effect of the three operators together, a specific schema S multiplies a certain number of copies in the next generation according to the following equation:

$$r(S, t+1) \geq r(S, t) \cdot \frac{f(S,t)}{\overline{f}}[1 - p_c \cdot \frac{\delta S}{l-1} - o(S)\dot{p}_m] \tag{4.8}$$

This development, though simple, is so important that it is given a special name: the *Schema Theorem*. An application follows for clarifying purposes and as an introduction to the ANN model application.

4.5 Application example

Prior to presenting the application of the GA to the ANN model, a demonstration of its performance is presented through a relatively simple example. This example was introduced to observe the application of the theory and principles presented above

through the GA's searching method, and because the ANN model's parameters to be defined by the GA form a multi-dimensional space difficult to visualise.

The example presented below represents an excellent way to prove the GAs effectiveness. The Eq. 4.9 plotted in Fig. 4.2 is a three-dimension surface with two independent variables and multiple local optima.

$$f(x,y) = \frac{sin^2(x-y)sin^2(x+y)}{\sqrt{2x^2+y^2}} \tag{4.9}$$

This equation was limited to the range $0.1 \leq x, y \leq 10$ on 0.1 increments, giving a search space $S = 10{,}000$ points. The maximum value $F(x_{max}, y_{max})$ is $F(0.1, 1.4) = 0.6565$. The several parameters to be considered to run a genetic algorithm are described in table 4.1:

Table 4.1: Main parameters in a genetic algorithm

Parameter	Description	Applied values
N_{gen}	number of generations allowed	5,000 (maximum)
N_{pop}	population size	8, 16
Elitism	the fittest individual is maintain	Yes
P_{cross}	cross probability	0.8
P_o	mutation probability	0.01-0.09

These parameters have been selected from the literature and allow the performing of tests to examine the difference on choosing values from several options. The initial population was 16 and randomly generated. Fig. 4.3 shows the contour map of Eq. 4.9 with the points explored by the GA in one of the runs. It can be seen how the individuals (represented here by each point) were concentrated around the best value as a consequence of multiplication of the fittest on each generation.

The population size has a marked influence on the algorithm's performance. Large population sizes require more computation time but they can speed con-

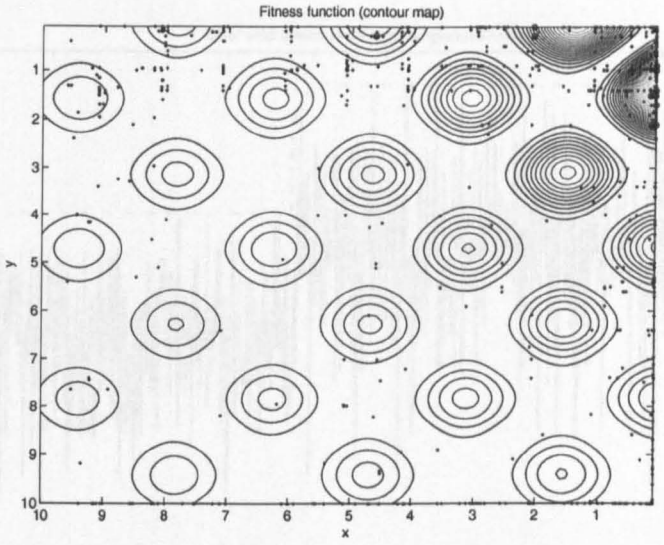


Figure 4.3: Contour map showing the points on the search space.

vergence. Graphs 4.4 and 4.5 present the GA behaviour with all the parameters identical other than the population size. The best individual is represented by the doted line.

The first test has been performed using 8 individuals. It required above 400 generations to find the best fitness value. The plot showing the best and mean fitness versus the number of generations is displayed in Fig. 4.4.

For the second test the number of individuals was 16. In this case the number of generations required to find the best fitness value was just above 60. Fig. 4.5 presents, likewise, the best and mean fitness against the number of generations.

To give a notion of the GAs efficacy and how they perform under different values of P_o, a test running the GA 50 times for every different value of P_o to produce the average and highest generation required to get the highest fitness value (0.6565) has been performed. Fig. 4.6 shows that the lowest number of generations required to get the best solution was at $P_o = 0.08$ using a population size of 16 individuals. In

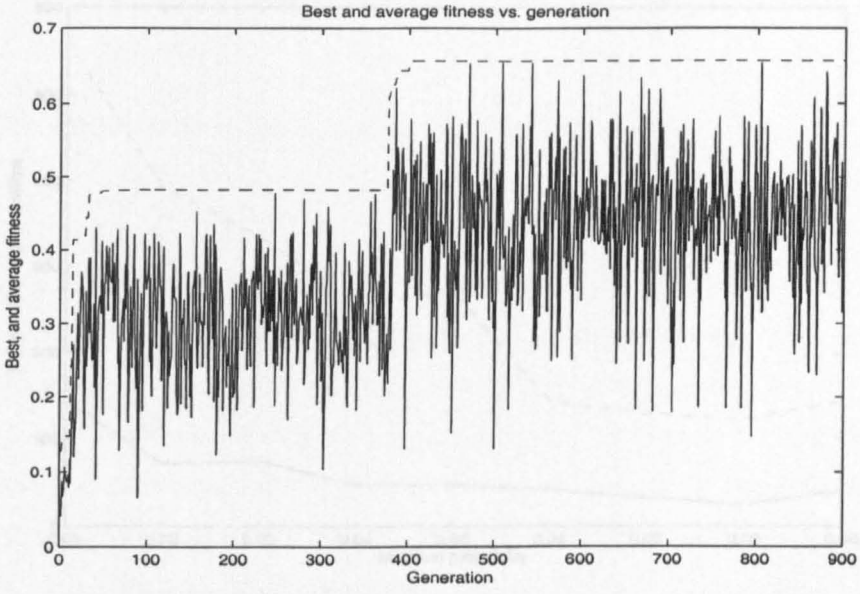


Figure 4.4: GA performance using 8 individuals and $P_o = 0.8$.

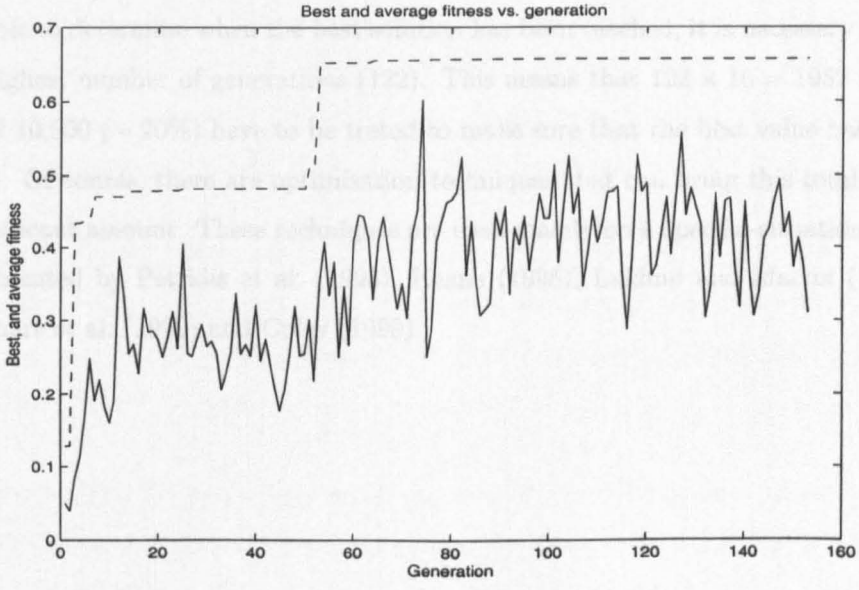


Figure 4.5: GA performance using 16 individuals and $P_o = 0.8$.

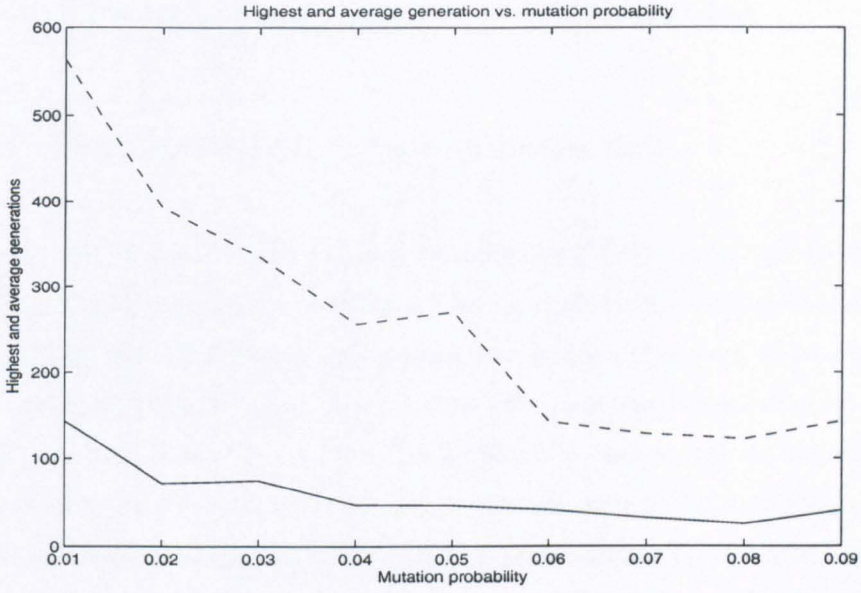


Figure 4.6: 50-run average for different P_o values.

average only 25.12 generations were needed to get the best solution. When it is not possible to determine when the best solution has been reached, it is necessary to use the highest number of generations (122). This means that $122 \times 16 = 1952$ points out of 10,000 ($\sim 20\%$) have to be tested to make sure that the best value has been found. Of course, there are optimisation techniques that can bring this total down a significant amount. These techniques are used mainly on a specific-situation basis as presented by Petridis et al. (1998), Keane (1995), Lakhmi and Martin (1999), Abrahart et al. (1999) and Coley (1999).

4.6 GAs application in the ANN model

4.6.1 Brue catchment using raingauge data

The GA used in the Artificial Neural Network (or ANN) model was focused on obtaining the best set of four parameters: number of layers, number of neurons in each layer, number of average-rainfall layers, and number of average data. Previous works have used GAs to optimised the first two parameters (see Abrahart et al. (1999)). In this thesis the use of a GA to find the best values for the last two parameters is presented. The average data was set between 20 and 400 previous rainfall values in increments of 10. When using two average values the second one was set at half the amount of the previous one.

The implementation of a GA for "tuning" the ANN model was based on the difficulty to find the best value for every parameter leaving them all but one fixed and, with a calculus-based method determine the optimum value for this parameter. In searching the best value for the next parameter the previous parameter's optimum value could not be used in this new search. This means that the search space is a multiple-optima one and is propitious for applying a GA.

The software program running the GA model has been written in C++ ANSI standard to ensure its portability (Liberty and Horvath, 2000). Most of the tests have been done under Linux because its processor management allows faster program runs than under Windows.

There were several criteria considered in order to delimit the parameter-value ranges: literature examples (Maier and Dandy, 2000; Fine, 1999; Bernard et al., 1994), computational cost, and tests with extreme parameter values. Table 4.2 shows the value range for every parameter:

Table 4.2: ANN parameters range used by the GA

Parameter	Description	Range values
N_l	number of layers	3-5
N_n	number of neurons on each layer	4-26 (in increments of 2)
A_l	average-rainfall input	0-2
N_a	number of average-flow data	20-400 (in increments of 10)

All these parameters represented a search space $I = (N_n(max)^{N_l(1)} + N_n(max)^{N_l(2)} + N_n(max)^{N_l(3)}) \times [N_a(max - min + 1)/10]^{A_l(max)}$ or, in figures $I = (12^3 + 12^4 + 12^5) \times 39 \times 2 = 21,161,088$. Therefore the GA tested just a small fraction of it.

The GA parameters used in the ANN model were obtained from several tests and, as it was described previously, by literature and experience from previous works. Table 4.3 presents the values employed in the GA for the runs.

Table 4.3: GA parameters used in tuning the ANN model

Parameter	Description	Applied values
N_{gen}	number of generations allowed	50
N_{pop}	population size	40
Elitism	the fittest individual is maintain	Yes
P_{cross}	cross probability	0.8
P_o	mutation probability	0.08

Figure 4.7 presents the lowest and mean error for one of the GA runs performed to tune the ANN model. The best individual after 50 generations was formed by 4 layers (10,8,10,1) and one average-data input: [-80, 1].

For a second run represented in Fig. 4.8 the best individual after 50 generations was formed likewise by 4 layers (10, 8, 8, 1) and one average input: [-100, -1].

From the two sets or 'individuals' the second one was selected. This choice was based mainly on the principle of parsimony. This second individual had a lower

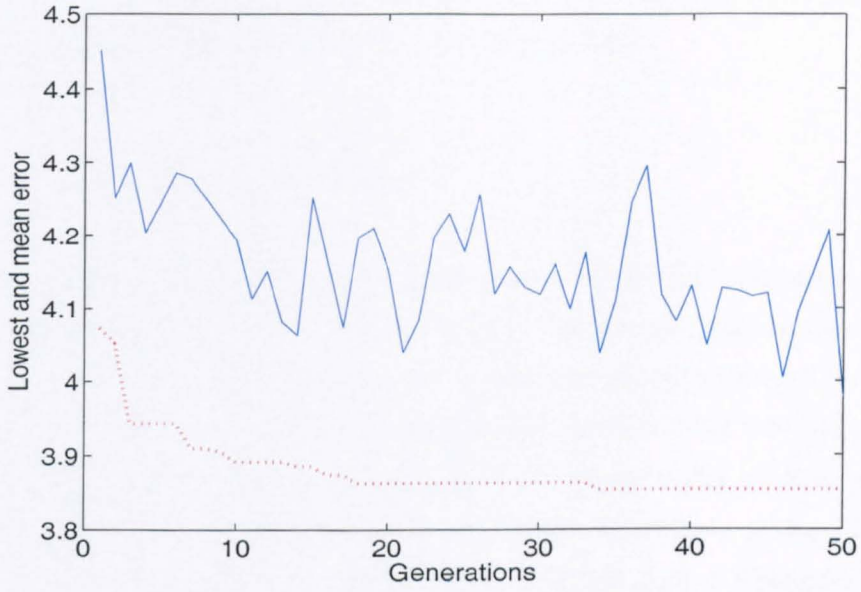


Figure 4.7: Lowest and mean error for a GA run used for the Brue catchment.

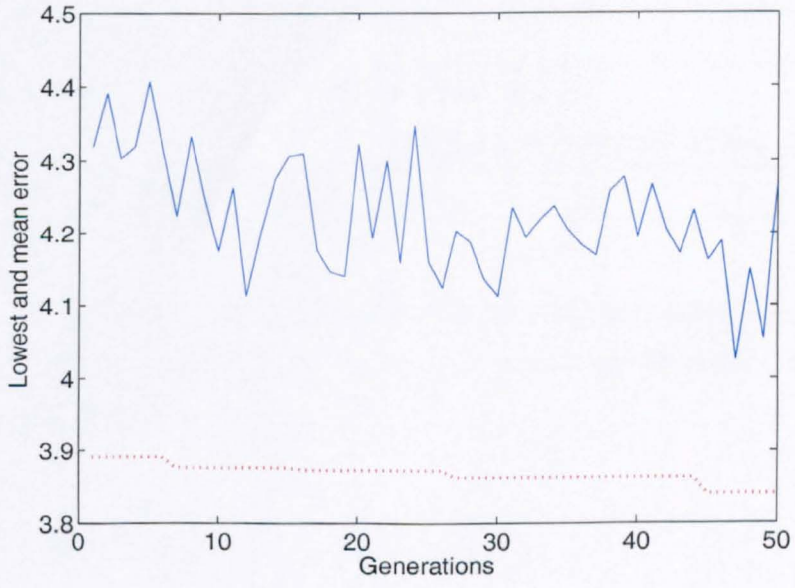


Figure 4.8: Best and average fitness for a GA run.

error: 3.8392 versus 3.8516 from the first individual, but due to the error variations in its simulations, this small difference won't be significant.

4.6.2 Bird Creek catchment

For the Bird Creek catchment the searching range of the ANN architecture was left equal to the range of the Brue catchment. The change that has been made was to discard the previous average-flow. It was done because the data from this catchment has a 6-hour time step. It means that the GA could test ANNs with previous rainfall values up to 156 hours before: 26 neurons in the input layer, each one representing 6-hour rainfall estimates. The range for previous average-flow for the Brue catchment model was only 100 hours (400 time steps ÷ 4 steps/hr). The set of values the GA worked with is presented in Table 4.4.

Table 4.4: Range of the ANN model's parameters for the Bird Creek catchment

Parameter	Range
number of layers N_l	3-5
number of neurons on each layer N_n	1 in last layer 4-26 (in increments of 2) in rest of layers
previous average-rainfall inputs A_l	None

The best parameter set identified by the GA was a four layer architecture formed by 8 input neurons, and 10 and 6 neurons in the second and third layers (first and second hidden layers).

4.6.3 River Glen

The river Glen is a tributary of the river Welland and both rivers form part of the Welland and Glen cathcment which is described in *Chapter* 6. Fig. 4.9 presents

a schematic diagram with the location of the points used in the ANN model to simulate the river.

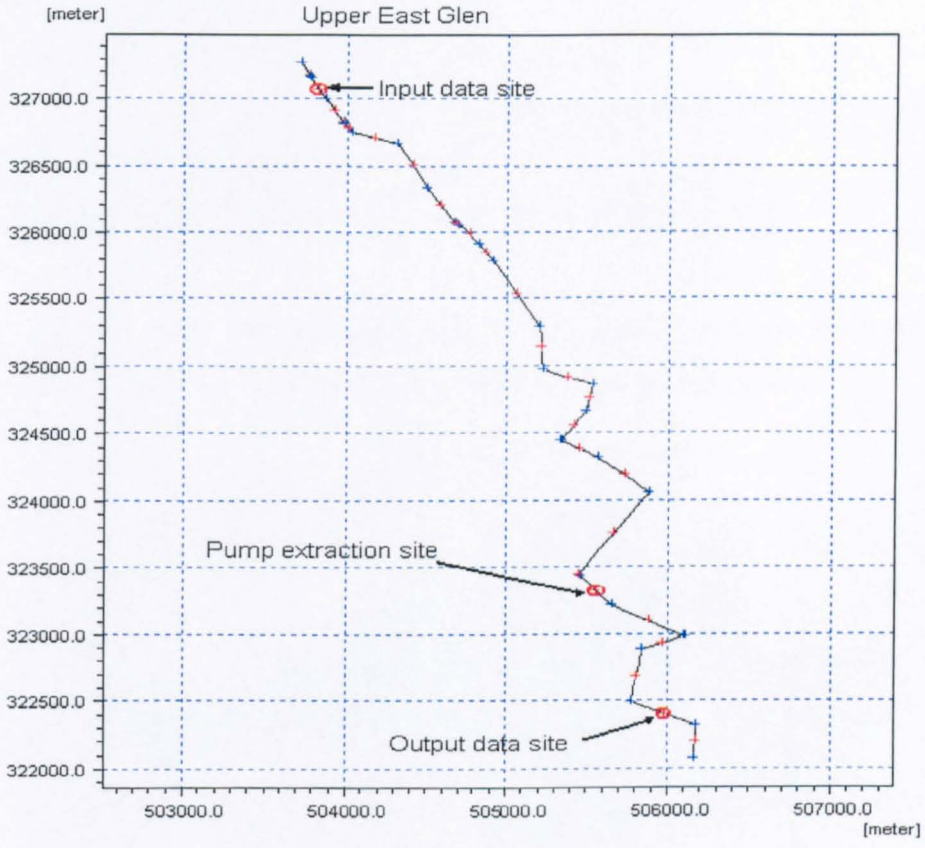


Figure 4.9: Schematic representation of the river Glen.

Modelling the Glen river meant using only flow data as input and output. The ranges of the parameters the GA searched within were almost the same to those used in the Brue catchment, the only difference was the shorter number of average-flow data. The set of values is displayed in Table 4.5.

The use of flow as input to the ANN model meant that the use of the filter in the signal preprocessing was not necessary.

The GA found out, for this river, an architecture of only 3 layers [6-16-1] and

Table 4.5: Range of the ANN model's parameters for the Glen river

Parameter	Range
number of layers N_l	3-5
number of neurons on each layer N_n	1 in last layer 4-26 (in increments of 2) in rest of layers
previous average-flow inputs A_l	0-2
number of average-flow data N_a	20-100 (in increments of 20)

1 average-input of 60 average-rainfall values. To visualise the search space the GA worked with, a three-dimension error surface is plotted in Fig. 4.10.

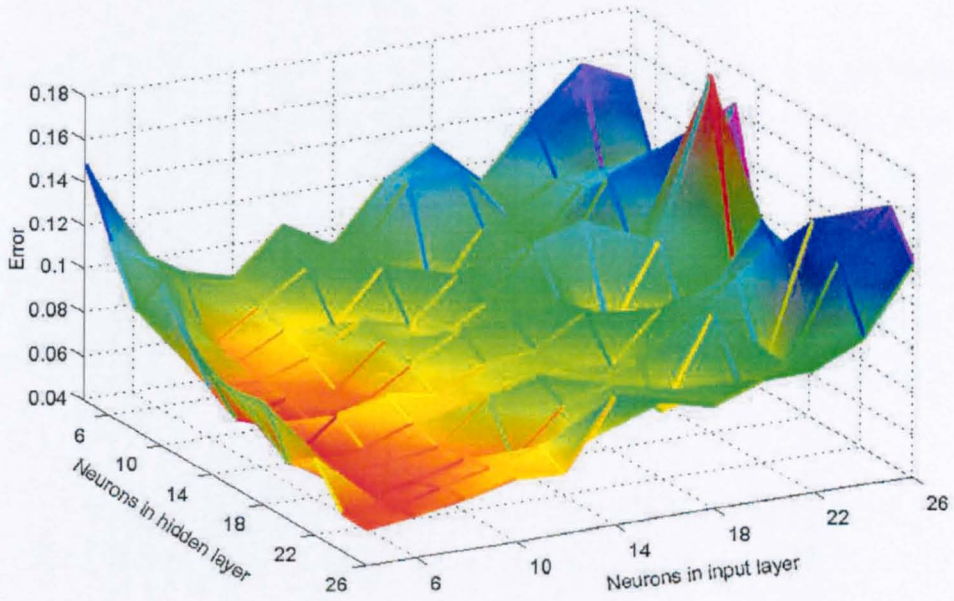


Figure 4.10: Error surface of the ANN modelling the Glen river.

This error surface is the 5-average of 5 runs in order to reduce local variations. The color gradation displays a redder colour for lower error values and a bluer colour for higher error values. Several local minima can be seen in the region where the selected architecture was selected.

4.6.4 Pump extraction

For the pump extraction, because this test was implemented in the river Glen model, the ANN model's architecture and parameters were left the same but added just one input. This input was added for testing the model under the condition of adding independent extraction rate. For the second condition studied the ANN model was exactly the same. The description of this analysis is presented in *Chapter* 6. The location of the extraction site in the river Glen is shown in Fig. 4.9.

4.6.5 River junction

The simulation of a river junction has been carried out using data from the Welland and Glen model. Fig. 4.11 shows the location of the junction and a detailed description is presented in *Chapter* 6. In this case there were two different flow series to be used as inputs for the ANN model. From the results obtained for the river Glen, the inputs were left fixed with values, for each flow series, of 6 inputs. This choice was taken to reduce the search space of the GA. The number of neurons in the hidden layers was changed to be between 12 and 28. All the parameters are presented in Table 4.6.

For this case, the final architecture selected by the GA was [12-28-28-1] and one previous average-flow input with 80 average data.

4.6.6 Brue catchment using radar data

The analysis of the ANN model using weather radar data was performed under three different implementations named: *average values*, *matrix and average values*, and *matrices*.

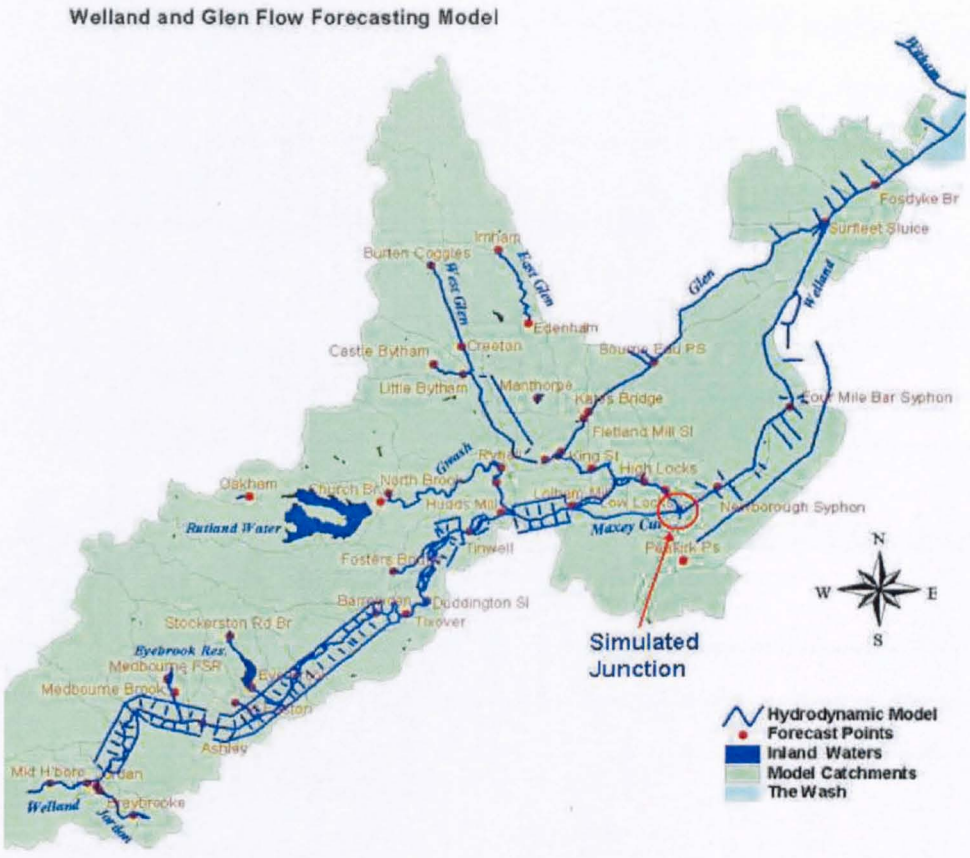


Figure 4.11: Welland and Glen catchment showing the location of the junction.

The *average values* implementation consisted, as indicated by its name, of the average of the 14 separate points from the radar grid overlapping the Brue catchment. This average series was handled as the raingauge rainfall series and, hence, the ANN model was the same as the one used for rainfall data (see section 4.6.1).

For the *matrix and average values* implementation, where there were 14 inputs coming from the radar grid, the input layer was fixed at 20 neurons. The inner layers' range was changed to a range from 12 to 28 possible neurons in increments of 2. The range for the number of previous average-flow inputs and number of average data was left equal to those employed for the Brue catchment using raingauge data. Table 4.7 presents all the ranges the GA utilised.

Table 4.6: Range of the ANN model's parameters for the river junction

Parameter	Range
number of layers N_l	3-5
number of neurons on each layer N_n	12 in first layer 1 in last layer 12-28 (in increments of 2) in hidden layers
previous average-flow inputs A_l	0-2
number of average data N_a	20-100 (in increments of 20)

Table 4.7: Range of the ANN model's parameters for the Brue catchment using matrix and average values

Parameter	Range
number of layers N_l	3-5
number of neurons on each layer N_n	20 in first layer 1 in last layer 4 - 26 (in increments of 2) in hidden layers
previous average-rainfall inputs A_l	0-2
number of average data N_a	20 - 100 (in increments of 20)

The GA search found an optimum architecture of 3 layers: 20-28-1 with one input for the previous average-rainfall consisting of 90 average-rainfall data.

For the third implementation, *matrices*, 5 matrices have been used, each one of 14 values, which meant 70 inputs plus an extra input for the moving average-rainfall. According to the increase in neurons for the first layer, the neuron range for the hidden layers has been increased to between 44 and 76 neurons. The complete list of parameter ranges is shown in Table 4.8.

Table 4.8: Range of the ANN model's parameters for the Brue catchment using matrices

Parameter	Range
number of layers N_l	3-5
number of neurons on each layer N_n	71 in first layer 1 in last layer 44-76 (in increments of 24) in hidden layers
previous average-rainfall inputs A_l	0-2
number of average data N_a	20-100 (in increments of 20)

The architecture determined by the GA was: 71-52-1 with one input for the previous average-rainfall consisting of 80 average-rainfall data.

4.7 Discussion

Building up an ANN model is not a straight forward task because if the model is run varying the values of one parameter and after all the values are tested the best one is chosen among them it does not mean the best option after working with another parameter. It means that, as it is seen in Fig. 4.10, the error surface is formed by multiple local minima. Thus the Genetic Algorithms were an invaluable tool for this purpose. They were used to define the ANN model's architecture even with the risk of being biased due to the multiple local minima. Abrahart et al. (1999) compared four design tool procedures to create improved neural network architectures for forecasting runoff, among them, a GA. In their conclusions they highlighted the fact that there is still no reliable scoring system that overcomes the difficulties of measuring peaks and troughs, or performs event-based separation of appropriate statistical descriptors.

Training a Neural Network demands a certain amount of computational time

that, understandably, varies with the processor employed. In this thesis three different processors were used: Intel Pentium III at 800 MHz, AMD Athlon at 2.4 GHz and Intel Pentium IV at 3.06 GHz. Variation in time was almost a simple relationship of processor speed, independently of memory RAM, cache, hard drive or any other feature. It took around one minute and 15 seconds to train a single neural network using the Pentium III processor and, around 15 seconds using the Pentium IV. Because the GA required training at least 2000 different neural networks, the amount of time demanded, even using the fastest processor, was very high, at least 8 hours. Thus, applying GAs as part of the standard ANN model for specifying its parameters must be used only once for building up a model structure for every specific catchment and reducing search space as much as possible, in order to reduce the computational time.

Chapter 5

Exploration of ANN in using weather radar rainfall data

> In later years, radar has been added to the long list of meteorological equipment. It provides an easy means of locating the radiosonde balloons and is much used in tracking thunderstorms, tornadoes, and hurricanes.
>
> Sverre Petterssen, 1958

5.1 Introduction

What was seen as a frequent interference signal on the early days of radar became, soon after, an independent research area: radar hydrology. For the origins of meteorological weather radars one must look at the invention of the cavity magnetron. Radars were already in operation, but their wavelengths (on the range of 66 and 133cm) were not short enough to detect atmospheric phenomena, (Atlas and Ulbrich, 1990). The cavity magnetron, developed by Boot and Randall at the University of

Birmingham in 1940, was able to generate short pulses at very short (a few centimeters) wavelengths (Austin, 1998). Displays using the new instrument showed, among others, an interference signal seemed to be associated with rain. This phenomenon was initially investigated by military groups in the UK, Canada, and the US, starting the development that lead to the meteorological field it is today.

This chapter is not intended to provide a comprehensive review of radar technology which have been detailed elsewhere (Cluckie et al., 2000; Collier, 1996; Tilford, 1992; Battan, 1973; Cole, 1992). Instead, a summary of the background and information necessary to gain an understanding of radar principles, basic theory, radar classification and problems associated with radars is given. The chapter continues by presenting the application of radar rainfall data in the ANN model. The radar data used in this experiment is spatially (geographically) distributed, generated by horizontal scanning round the radar. The image obtained is a grid where every point represents a rainfall estimation at a specific time. The ANN model is fed with the points overlapping the catchment under study. Also, these points are spatially averaged to run the ANN model with two different data structures to compare their performances.

5.2 Background

Radar, by definition, is an electromagnetic system which performs the tasks of **RA**dio **D**etection **A**nd **R**anging on distant target objects. The more formal definition of radar, as expressed by Battan (1973) is: 'the art of detecting by means of radio echoes the presence of objects, determining their direction and range, recognising their character and employing the data thus obtained '.

It is in 1936 when a team lead by Watson-Watt developed the radar. At the beginning of the Second World War radars were used by several countries, UK, US

and Germany among them. These first radars operated in the range of 200 and 400 MHz, Fletcher (1990). It was until the Boot and Randall invention, with the ability of the magnetron to generate microwaves above the Gigahertz in frequency, that the possibility to detect hydrometeors[1] was first discovered. Since then, the use of radars for meteorological application has expanded rapidly (Collier, 1996). Ligda (1951) as referred to by Atlas and Ulbrich (1990), stated that the first time rain was observed by radar occurred on the 20th of February 1941, somewhere on the south coast of England. Nevertheless their potential and possibilities, weather radars were not so rapidly adopted by weather forecasters, as demonstrated in Sutton (1961). In this book the author, Director-General of the Meteorological Office in the UK at the time of its publication, does not mention radar applications at all.

Weather radars, as any other radar, send electromagnetic waves to a target area and, if something crosses the signal beam, receive a small fraction, in terms of power, of the original signal. This reflected signal or 'reflectivity'must be expressed in terms of rainfall. Several options have been proposed up to date but, among them, the Marshall-Palmer relationship (Eq. 5.12) is still one of the most important equations in radar meteorology. This equation is the research product of a group established in 1943 and led by Stuart Marshall (the Stormy Weather Radar Group) in Canada.

David Atlas in 1945 in the United States and R. F. Jones in 1946 in England set up research and operational centres of radar meteorology. During the last two decades several countries extended their national meteorological radar network including USA, UK, Germany, Canada, Italy, New Zealand, China, India, Japan and Brazil amongst others. For a review of historic development of weather radar in different countries see Swingle (1990); Probert-Jones (1990); Douglas (1990); Kodaira and Aoyagi (1990).

In the UK at a practical level, several X-band (3cm wavelength) radars were placed at various locations on major air routes and one on the roof of the Air

[1]any form of precipitation (rain, snow, hail, etc.)

Ministry in London in the 50's (Collinge and Kirby, 1987). In the mid 1960's the major movement is made from rainfall detection to rainfall measurement through the Dee Weather Radar Project (DWRP). DWRP resulted in the installation of an S-band (10cm wavelength) at Llandegla, Wales. In 1973 the radar was converted to a C-band (5cm wavelength).

Although research using weather radar in the UK continued since its discovery in the 40's, it was not until the mid-70's that hardware advances enabled data to be processed in real-time, transmitted from the site and displayed remotely.

A network of twelve, C-band radars were suggested as a national weather radar network by a group of experts to cover the British Isles (Hajjam, 1997). Following the DWRP recommendations, a Plessey C-band radar was established at Hameldon Hill in May 1980 by a consortium including the Meteorological Office and the North West Water Authority.

There are currently some 15 C-band weather radars (plus an S-band device at Shannon Airport) networked and in operation in the UK and Ireland. Their distribution and coverage is shown in Fig. 5.1. An example of rainfall information from the Meteorological Office covering all the UK, Ireland, and Jersey is shown in Fig. 5.2.

The development of weather radars and the advent of satellite sensing provide the appropriate sources of data for short period mesoscale rainfall forecasting. In the UK these two data sources were combined through the Short Period Weather Forecasting Pilot Project (SPWFPP) and its successor, Forecasting Rainfall Optimised using New Techniques of Interactively Enhanced Rainfall and Satellite (FRONTIERS) (Hajjam, 1997).

Radar data are first collected in polar co-ordinates and are then converted to Cartesian coordinates at site of the original radar. Network data are only available

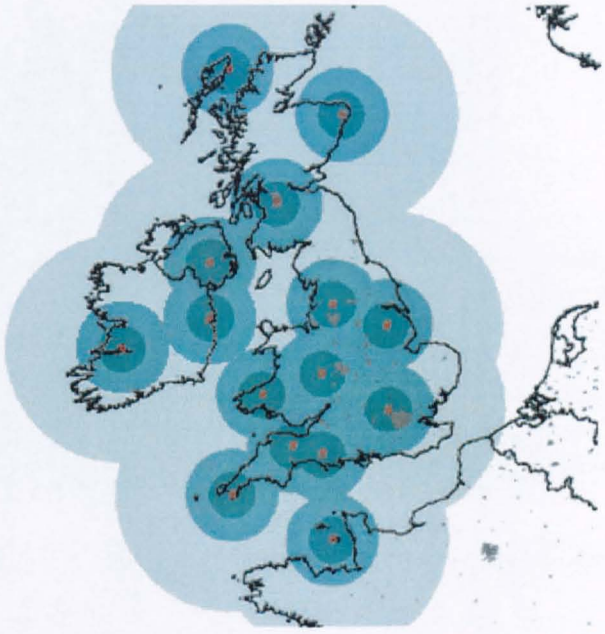


Figure 5.1: Radar network coverage in the UK provided by the Met Office.

from Meteorological Office headquarters at Exeter. In order to create the national images at Exeter, radar data are composed with data from UK, Jersey and Ireland radars. These data are also composed with European radar data to produce COST73 images and with Meteosat data from FRONTIERS information.

5.3 Basic Theory

Radars provide indirect measurement of rainfall by emitting electromagnetic energy and observing the nature of the energy reflected back. As shown in Fig. (5.3) radars utilise the microwave range of frequency (f) spectrum. For meteorological purposes radars are often described in terms of their wavelength (λ) rather than the frequency. These two parameters are related by $c = \lambda f$, where c is speed of light. In general, the smaller the size of the particles, the shorter the wavelength required

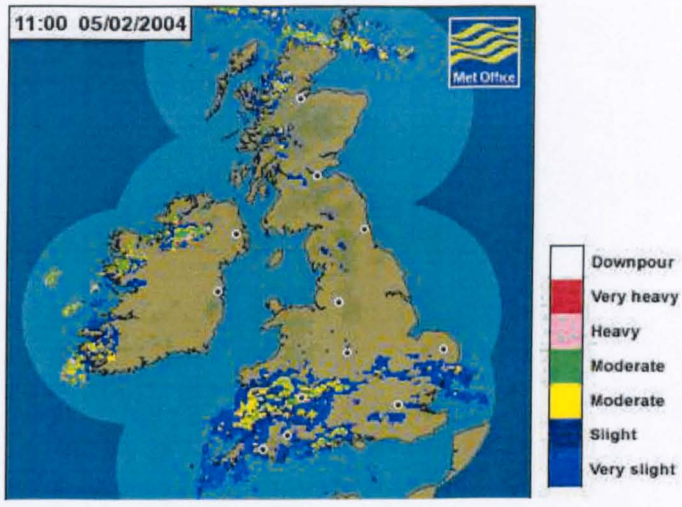


Figure 5.2: Rainfall radar image provided by the Met Office.

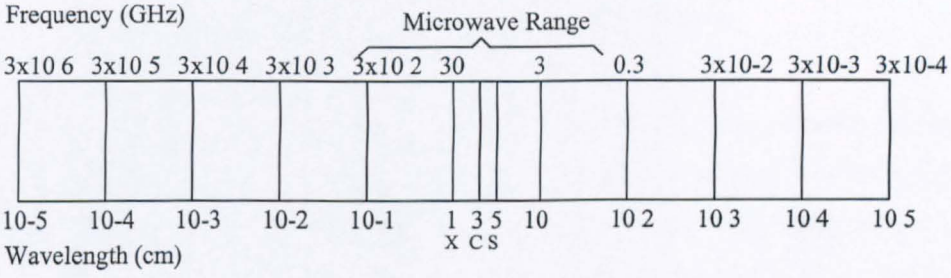


Figure 5.3: The Electromagnetic spectrum frequencies of 3×10^6 to 3×10^{-4} GHz.

to detect those particles. In meteorology most often the C-band (5cm) or S-band (10cm) wavelength are used.

A scheme of the hardware of weather radar operation is shown in Fig. 5.4. A transmitter emits an electromagnetic pulse at a known and given frequency. This energy is propagated through an antenna. When it intercepts an object (target), the pulse is partially reflected and returns to the same antenna. The antenna then directs any returned echoes to a receiver by means of a transmit-receive switch.

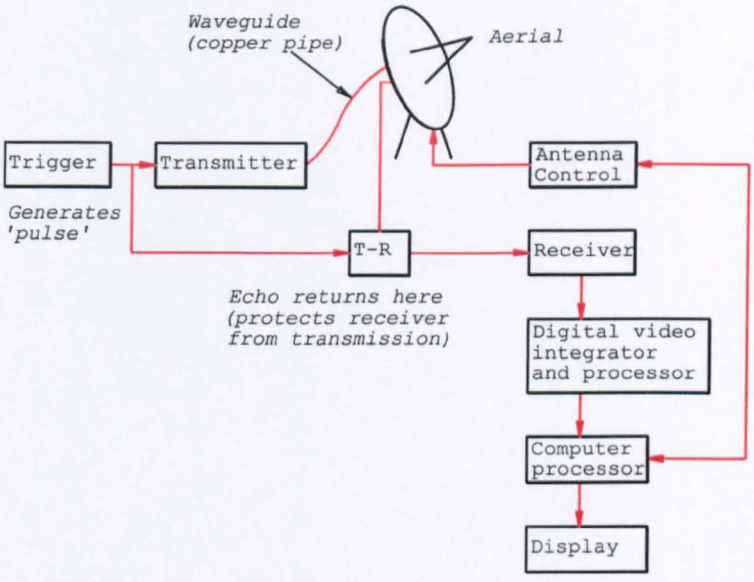


Figure 5.4: Weather radar operation scheme.

The radar receiver directs the received echo to a so-called digital video processor and computer processor. Through these, the signal is converted into meaningful information which can be displayed on a computer screen. The distance from the object also is determined by measuring the time interval between the transmission of the radio energy and the reception of the reflected signal.

As indicated by Hajjam (1997) the power received, p_r is a function of the transmitter system, the propagation path from the transmitter system to the target, the propagation path from the target to the receiving system, and the receiving system. The radar equation may be summarised as follows:

$$\overline{p}_r = ABCDE\Sigma\sigma \tag{5.1}$$

where the terms represent:

- A = transmitting system = $p_t G_t / L_t$
- B = propagating medium in the direction of the target= $1/(4\pi r_t^2 L_{mt})$
- C = propagating medium in the direction of the receiver= $1/(4\pi r^2 L_{mr})$
- D = receiving system = $G_r \lambda^2/(4\pi L_r)$
- E = polarisation effects = $1/L_p$
- σ = radar cross-section (also called back-scattering cross-section)

and the parameters are defined as:

- p_t = transmitter power in watts;
- G_t = gain of the transmitting antenna in the direction of the target (the amount that antenna focusing increases power);
- L_t = numerical factor to account for losses in the transmitting system;
- L_r = a similar factor for the receiving system;
- L_{mt}, L_{mr} = numerical factors which allow the propagating medium to have loss;
- r_t = range between the transmitting antenna and the target;
- r = range between the target and receiving antenna;
- G_r = gain of the receiving antenna in the direction of the target;
- λ = radar wavelength;
- L_p = numerical factor to account for polarisation losses

Usually it can be assumed that: $G_t = G_r = G$ and $r_t = r$ therefore, the Eq. 5.1 can be written as:

$$\overline{p}_r = \frac{p_t G^2 \lambda^2}{(4\pi)^3 r^4} \Sigma\sigma \tag{5.2}$$

The back-scattering cross-section, σ, is defined as the area which would intercept sufficient power from the transmitted field to produce the gain echo at the receiving system by isotropic radiation. The amount of energy back-scattered from the hydrometeors depends upon the number of particles within the pulse volume of the radar beam, their size, composition, relative position, shape and orientation.

G. Mie, as cited by Battan (1973) expressed the back-scattering cross section σ as:

$$\sigma = \frac{\pi D^2}{\alpha^2} \left[\sum_{n=1}^{\infty} (-1)^n (2n+1)(a_n - b_n) \right]^2 \tag{5.3}$$

where D is the drop diameter, $\alpha = \pi D/\lambda$ and is called the electrical size, a_n and b_n are coefficients of the scattered field. If the drop diameter is small compared to the wavelength (i.e. $\alpha < 0.13$) then a simplification of Eq. (5.3) may be used in accordance with the Rayleigh scattering formula (this equation is referred to as the 'Raleigh approximation' of the backscattering cross section) as:

$$\sigma = \frac{\lambda^2 \alpha^6}{\pi} \left| \frac{m^2 - 1}{m^2 + 2} \right|^2 \tag{5.4}$$

where $m = n - ik$ (the complex index of refraction), n = the ordinary refractive index, and k = a coefficient of absorption. If $K = \frac{m^2-1}{m^2+2}$ then the Eq. 5.4 can be written as:

$$\sigma = \frac{\pi^5}{\lambda^4} |K|^2 D^6 \tag{5.5}$$

this shows that scattering cross-section is proportional to the sixth power of the drop-size diameter.

Probert-Jones (1962) expanded the Eq. 5.2 to allow for the beam shape and other factors as:

$$\overline{p}_r = \frac{p_t G^2 \lambda^2 \theta \phi h L}{512(2\ln 2)\pi^2 r^2} \times \frac{1}{\Delta v} \sum_{vol} \sigma \tag{5.6}$$

where L is the sum of all the losses which includes the attenuation by atmospheric gases, precipitation and the radome[2]; θ and ϕ are the vertical and horizontal beam widths and h is the pulse length, and Δv is the pulse volume. An idealised radar beam, of width θ and inclined at angle α under conditions of normal atmospheric propagation is shown in Fig. 5.5.

Substitution of Eq. 5.5 into 5.6 gives:

$$\overline{p}_r = \frac{p_t G^2 \theta \phi h L \pi^3}{512(2\ln 2) r^2 \lambda^2} |K|^2 \frac{1}{\Delta v} \sum_{vol} D^6 \tag{5.7}$$

If the radar constant (C) and radar reflectivity (Z) are assumed to be:

$$C = \frac{p_t G^2 \theta \phi h L \pi^3}{512(2\ln 2)\lambda^2} \tag{5.8}$$

$$Z = \frac{1}{\Delta v} \sum_{vol} D^6 \tag{5.9}$$

[2](radar dome) is a weatherproof enclosure used to protect an antenna

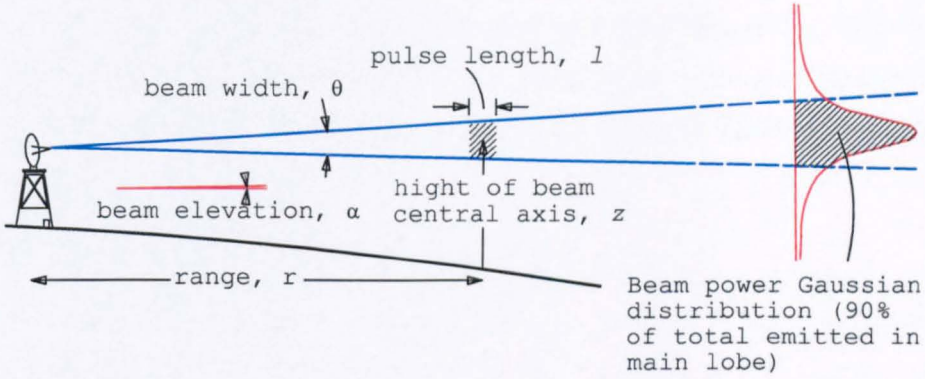


Figure 5.5: Geometry of an idealised pulse beam.

then the Eq. 5.7 can be written as:

$$\overline{p}_r = \frac{C|K|^2 Z}{r^2} \tag{5.10}$$

Equation 5.10 is subject to numerous assumptions which are listed by Collier (1996), the principal ones being that:

- Rayleigh scattering theory is applicable;
- the pulse volume is completely filled by randomly distributed precipitation particles;
- Z is uniform throughout the sampled pulse volume and is constant during the sampling interval;
- $|K|^2$ is the same for all the hydrometeors within the sample, i.e. they are either water drops or ice particles;
- absorption of the transmitted signal by ground clutter in the beam is negligible.

It should be emphasised that it is not practical to measure operationally $\sum D^6$, therefore Z (radar reflectivity) may be calculated according to Eq. 5.10 if the range r and the average received power, $\overline{p}_r$ are known. In order to calculate rainfall intensity Z, it is related empirically to the rainfall intensity by the $Z-R$ relationship as follows:

$$Z = aR^b \tag{5.11}$$

where R is the rainfall intensity and a and b are empirically derived constants. The appropriate units for Z and R are mm^6m^{-3} and $mmhr^{-1}$ respectively. The values of parameters a and b are a function of rainfall type, radar properties, and a range of values have been estimated for use in varying conditions. A wide range of values of a and b parameters have been identified by different researchers for different precipitation. Most widely used for homogeneous events is that of Marshall-Palmer as:

$$Z = 200R^{1.6} \tag{5.12}$$

There are a number of problems in using weather radars which are described in section 5.4.2.

5.4 Operation of weather radars

5.4.1 Radar information display types

The radar has two rotation modes - one that changes the angle of elevation (angle of the beam axis with respect to the ground), and the other (rotation around the

vertical) that changes the angle of azimuth (angle measured with respect to the north). This enables two types of scans: a series of two-dimensional images called **P**lan **P**osition **I**ndicator (PPI), which are scans around the vertical with the elevation angle fixed; and the **R**ange-**H**eight **I**ndicator (RHI), which are scans by varying the elevation (up-down direction) with the azimuth fixed. Radars fixed at a zenithal position are called Vertically Pointing Radars (VPR); their scans render **H**eight-**T**ime **I**mages (HTI). The diagrams of three types of scanning are presented in Fig. 5.6. The displays obtained by the three scanning types are colour-coded to denote intensity variations and examples of each one are displayed in Figs. 5.7 for a HTI, 5.8 for a RHI, and 5.9 for a PPI.

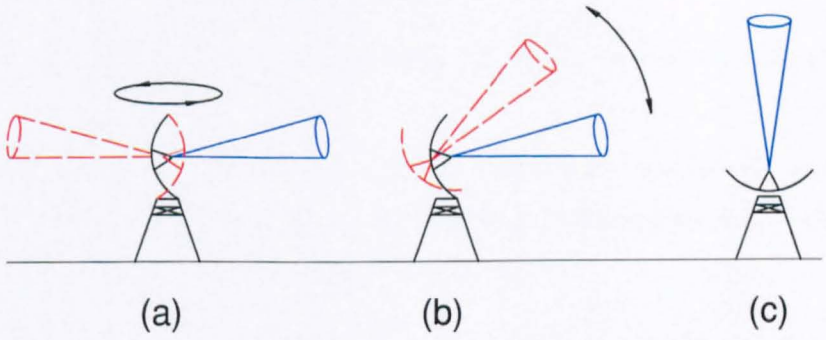


Figure 5.6: Scanning types employed in meteorology: (a) PPI, (b) RHI , (c) HTI.

The information display to be used in the ANN model is the PPI which covers geographically the catchment area under study.

5.4.2 Problems arising from the operation of radars

- **Ground clutter and occultation** Ground clutter is a persistent echo caused when the main part of the radar beam and/or the side lobes encounter ground

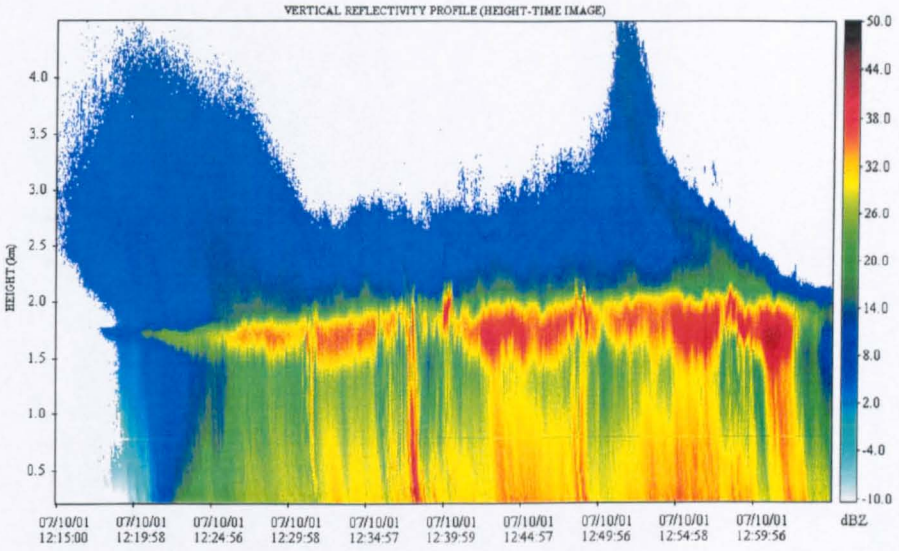


Figure 5.7: HTI generated by a Vertical Pointing Radar. Source: WEMRC (2004).

targets. Interception of the beam by the ground can cause, as well, occultation or screening of the main part of the beam such that only a fraction of the power targets the rain at longer ranges. See fig. 5.10.

- **Radome attenuation** The antenna of the radar is often housed in a radome made from rubberized material or fibreglass and either inflated or supported by aluminium strips. Attenuation of the radar beam is caused by precipitation wetting or clinging to the radome.

- **Anomalous propagation** The radar beam can be affected by abrupt changes in temperature or humidity, changing its normal path and then, targeting unexpected objectives as ground clutter.

- **Bright band** This is one of the most common and severe problems in measuring rainfall from radars. Bright band represents a zone where ice particles and snowflakes begin to melt down (the bright band is also known as melting layer). The reflectivity increases (approximately five times) when, due to the

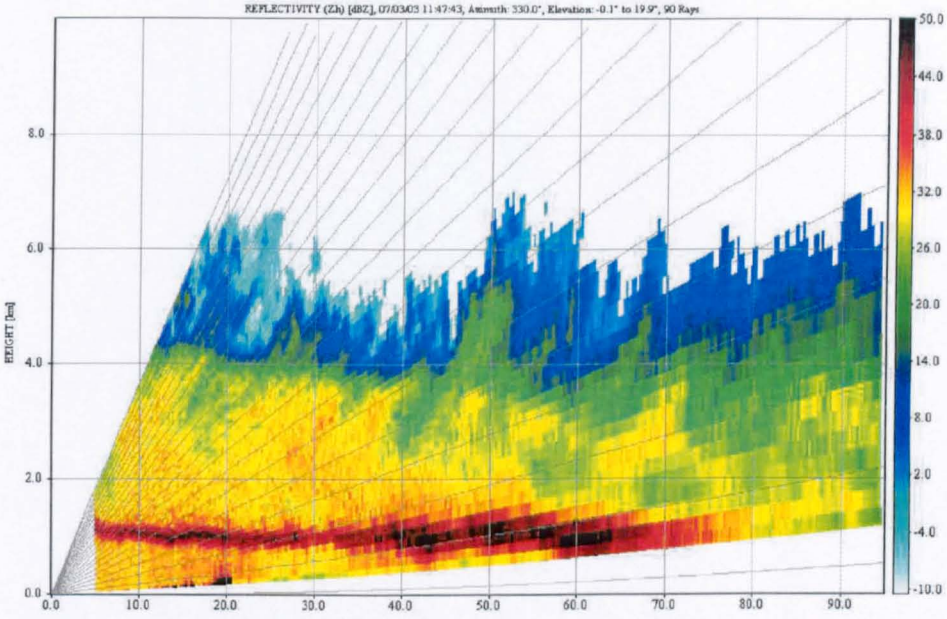


Figure 5.8: RHI generated by the Chilbolton radar. Source: BADC (2004).

increment in temperature, these particles are wrapped by a thin coating of water.

- **Z-R relationship** The values of a and b specified in Eq. (5.12) are defined for a set of specific atmospheric conditions. These parameters can vary considerably because they are highly determined by the size and distribution of particles within a given volume of the radar pulse which change dramatically either temporally or spatially.

- **Radar calibration** Radar hardware components (transmitter, receiver, etc.) stability and sensitivity calibration are a source of error that, with up-to-date components can be limited to within 2 dB or 36% error of rain rate (Joss and Waldvogel, 1990).

The radar hydrology community maintain their research effort on overcoming the above problems as seeing in Joss and Waldvogel (1990); Smith (1990); Illingworth

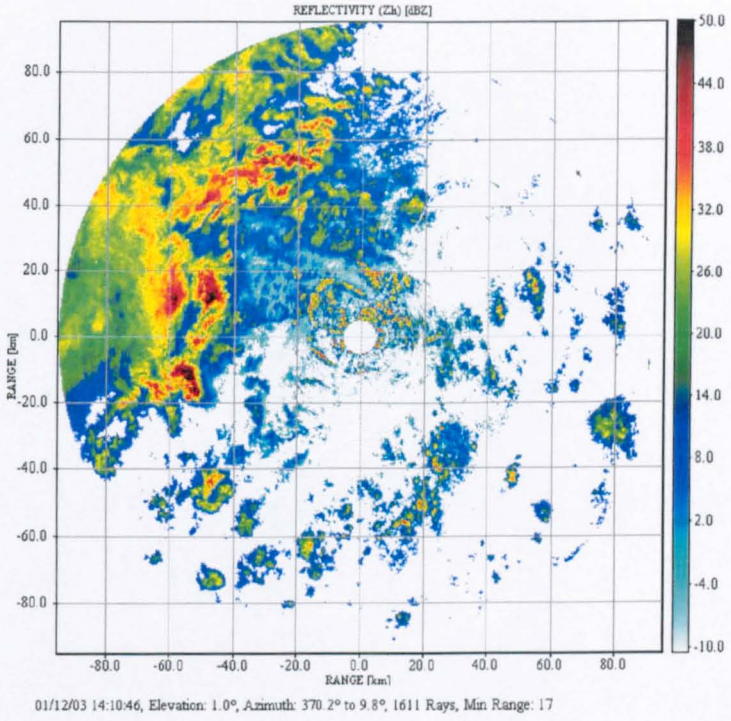


Figure 5.9: PPI generated by the Chilbolton radar. Source: BADC (2004).

et al. (2000). Harrison et al. (2000) examines the steps taken by The Meteorological Office, where the date used here is generated to address these problems.

The radar information to be applied to the ANN model spans 4 years: from 1995 to 98. The problems described above are intrinsic to this information and, the following tests intend to evaluate the extent to which the ANN model can deal with them.

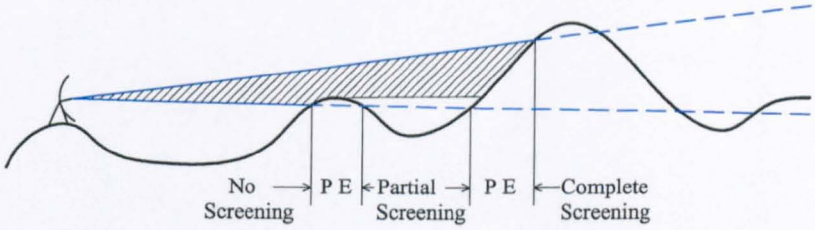


Figure 5.10: Illustration of the occurrence of screening and permanent echo (PE)(clutter) in a hilly region, from Collier (1996).

5.5 Runoff forecast by the ANN model using radar data

Rainfall estimations generated by weather radars, once they are processed, are made available by the Met Office, in the UK, every 15 minutes. The ANN model will be trained with these time steps of rainfall estimations to generate flow estimations. The process employed for such objective is described in the following sections and the results presented afterwards.

5.5.1 Introduction to the ANN experiment using radar data

The information used to perform this experiment was obtained, like one of the experiments in Chapter 3, by the HYREX project. This project generated rainfall estimates from raingauges and from weather radars. A network of raingauges were installed across the Brue catchment, in Somerset, South-West England. Radar rainfall data was obtained from continuously scanning C-band radars at Wardon Hill, 30 km south of the catchment, at Cobbacombe Cross, 70 km to the west, and and experimental Doppler dual-polarisation S-band radar at Chilbolton (Fig. 5.11).

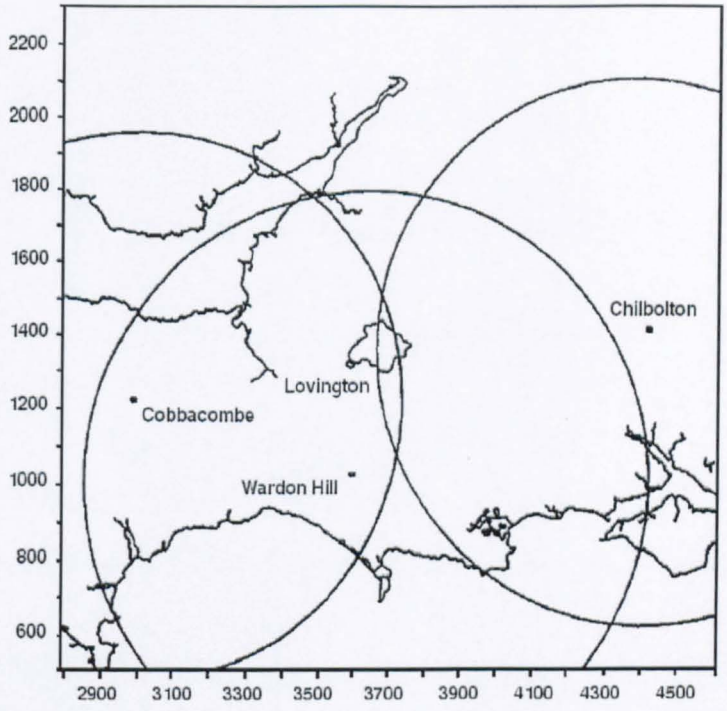


Figure 5.11: The scanning radars overlapping the Brue catchment gauged at Lovington.

The Brue catchment is overlapped by 14 of the 5 km squares which form the radar data grid displayed in Fig. 5.12.

This grid of radar data presents spatial and temporal variations. Fig. 5.13 shows hyetographs for squares D, G, and L (from grid displayed in Fig. 5.12) along the same time period: 17-Jan-1995 20:45 to 02-Feb-1995 07:15.

This hyetographs present a general similar patter with variations at some specific points. Part of the study was to compare the impact on the ANN model performance of both a distributed rainfall estimation and the average of this. Rainfall precipitation from the raingauge network and radar grid has been averaged to produced one-dimension rainfall series from each systems. Radar rainfall estimations are given in instantaneous values of mm/hr. Rain gauge data series, obtained like-

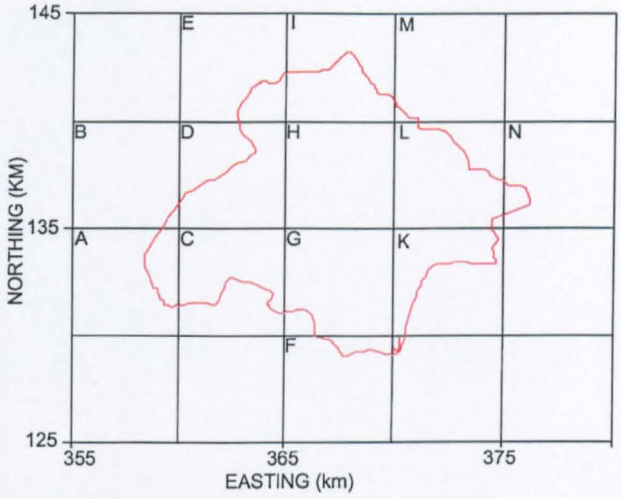


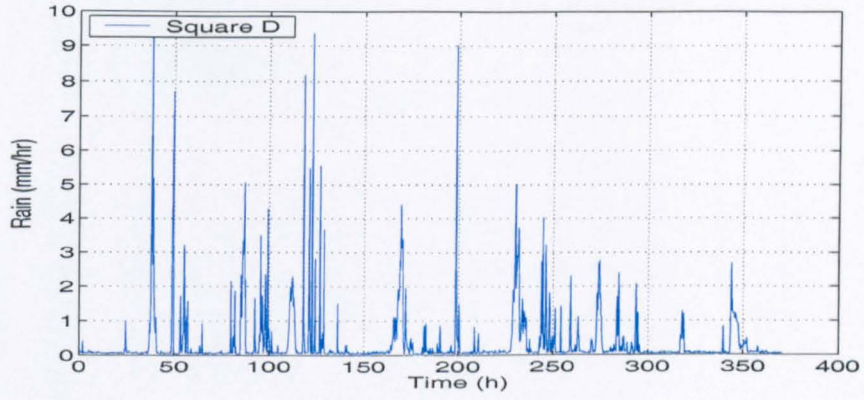
Figure 5.12: Radar grid covering the Brue catchment.

wise, every 15 minutes, are the accumulated rainfall along those 15 minutes; hence they were multiplied by 4 to present them in mm/hr. Fig. 5.14 compares the two average rainfall series; the period coincides with that of Fig. 5.13.

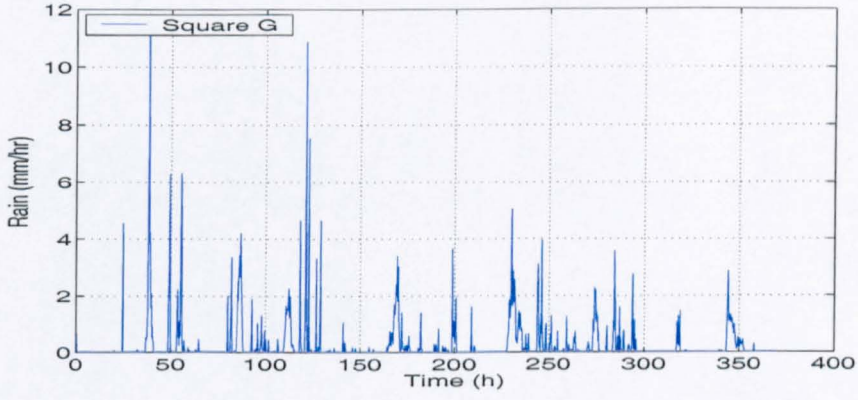
The cumulative rainfall from radar and raingauge is shown in Fig. 5.15. In this plot the gap between both cumulative series goes up to 12.3 mm. There is notoriously a tendency to estimate lower precipitation by the radar measurements during almost all the period. This bias would not affect the ANN model performance as long as it is consistent along the time due to the preprocessing of the signal (see Sec. 3.9).

Running the model with average rainfall data made it work like a lumped model; that is, the spatial (geographic) variations of rainfall values were just averaged. Fig. 5.16 plots a radar average rainfall series against the single values of each one of the points overlapping the Brue catchment. The period displayed covered from 05:15 to 21:30 on 08-Jan-1996.

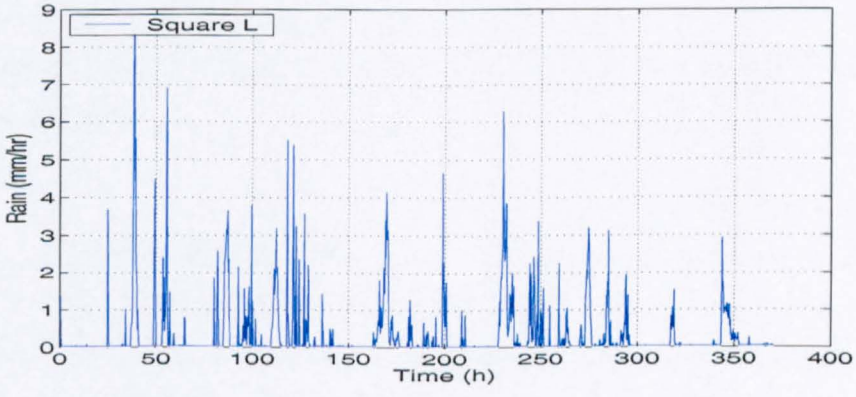
The radar rainfall series contained many short periods of missing data; this



(a)



(b)



(c)

Figure 5.13: Hyetographs for squares (a) D, (b) G, and (c) L.

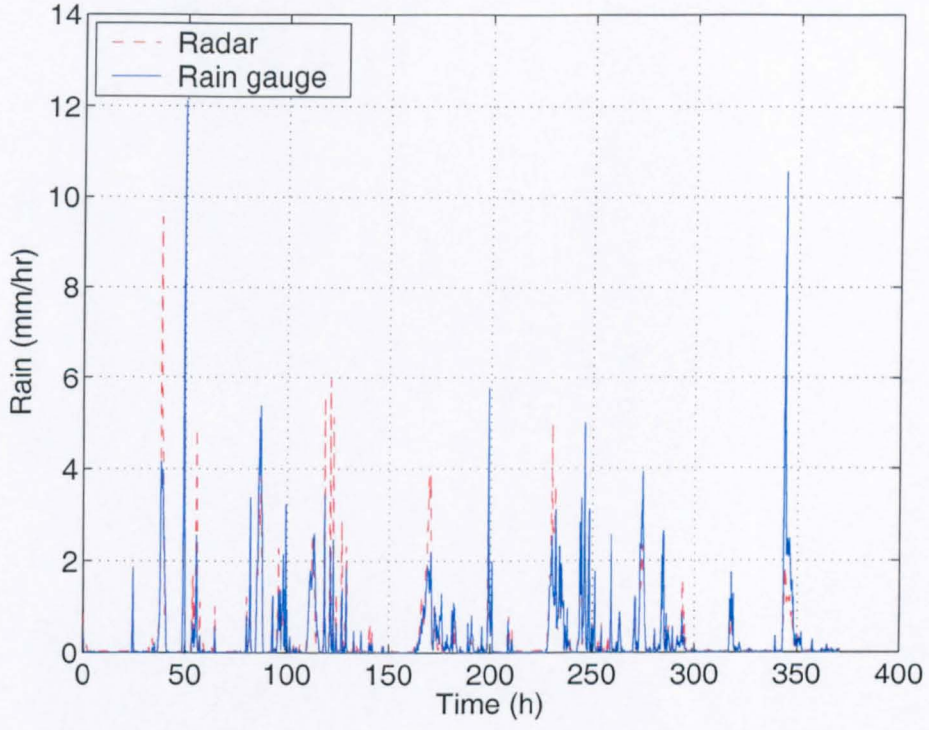


Figure 5.14: Hyetographs estimated by raingauge and radar.

situation limited the number of periods with enough values available to train and test the ANN model. Nevertheless, the tests allowed a proper comparison among the different options employed to run the model. These options are described in the following section.

5.5.2 Model description

The ANN model used to simulate the rainfall-runoff relationship was based on the model described in Sec. 3.16 for simulating the same catchment rainfall-runoff relationship. This time radar data were the information to work with; raingauge data were employed for comparison purpose.

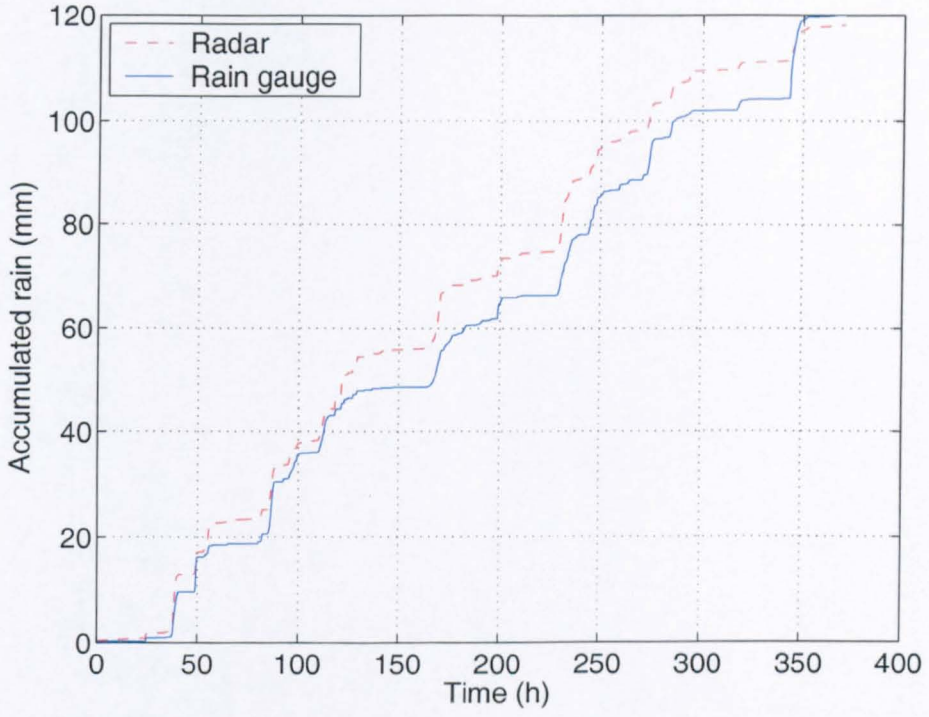


Figure 5.15: Cumulative hyetographs estimated by rain gauge and radar.

The rainfall data has been structured as a matrix of 14 columns (one for each 5km-square overlapping the Brue catchment) by n rows, where n is the number of data units in the selected period of time (every 15-minutes). These data series have been applied to the ANN model through three different implementations:

1. *Average values.* A one-dimension rainfall series. Every value in this series is a row-average or average of the 14 columns aligned on time. This implementation is similar to the raingauge series. The signal preprocessing is identical to the implementation applied in the case of the raingauge data. It included the IIR filtering with 18 coefficients, mean removal and resizing of the data series. An average of 80 previous values was added as one of the inputs to the ANN model.

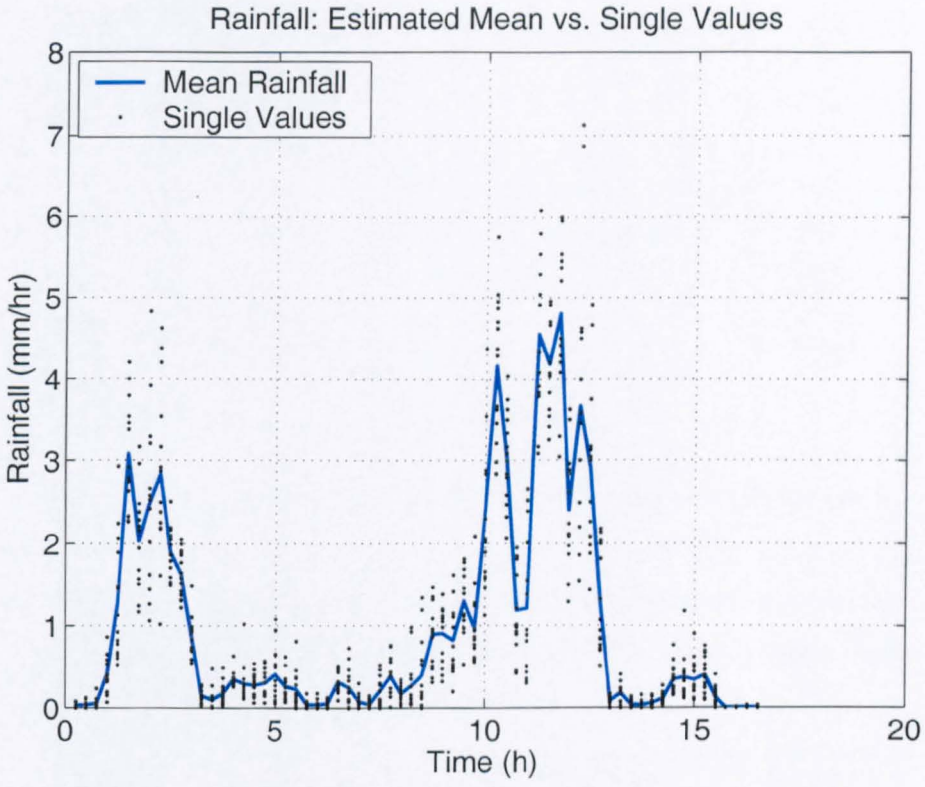


Figure 5.16: Radar average and single values of rainfall.

2. *Matrix and average values.* A matrix of distributed values at time t plus row-average values at times $t-1$, $t-2$, ... and $t-9$. Fig. 5.17 shows a graphical description of this implementation. The matrix displayed in Fig. 5.12, presented here at time t represents the 14 distributed values of rainfall estimation overlapping the Brue catchment. All these values were filtered along time with the previous values corresponding to the same spatial position, making the filtering work on data series like the three series presented in Fig. 5.13. The average of 80 previous average values was equally added to the model.

3. *Matrices.* The last data structure was a series of matrices at time t, $t-1$, $t-2$, $t-3$, and $t-4$. Fig. 5.18 presents a graphical description of this structure.

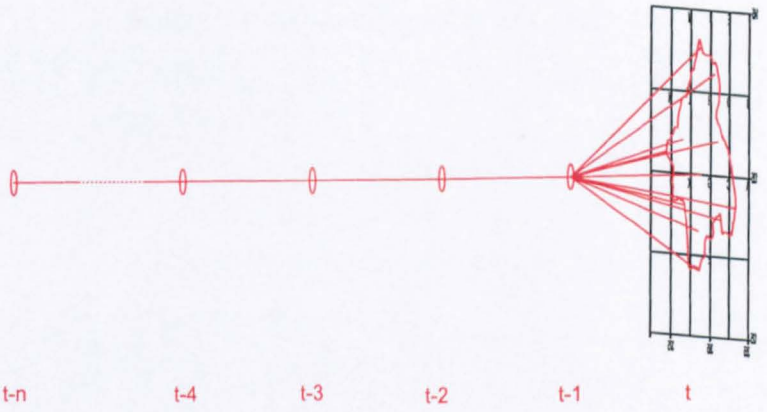


Figure 5.17: Data processing by the ANN model: *matrix and average values.*

In this case the filtering has been implemented individually on each one of the 14 points along time. Equally, the average of 80 previous average values was added as one of the inputs, to the model.

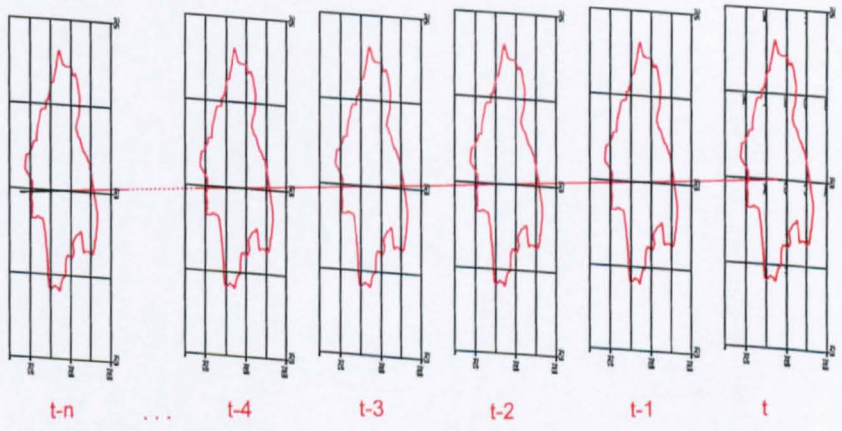


Figure 5.18: Data processing by the ANN model: *matrices.*

The model, though based on a previous model as indicated above, had some parameters that varied according to the data structure employed. Table 5.1 presents the ANN architecture.

Table 5.1: ANN architectures of the model

Parameter	Description	Value
N_l	number of layers	4 and 3
N_n	number of neurons on each layer	
	for average values (structure 1)	10 - 8 - 8 - 1
	for matrix and average values (structure 2)	20 - 16 - 1
	for matrices (structure 3)	71 - 44 - 1
A_l	average-previous flow values	1
N_a	number of average data	80

The number of layers and neurons in each layer for the model using data structures 2 and 3 were obtained, like the original model architecture running the data structure 1, using the genetic algorithm described in Chap. 4.

5.5.3 Results

The first data set to train and test the ANN model was from the winter months: February 1995 for training the model and January 1996 for testing it. Table 5.2 presents the exact dates and times applied to the model.

Table 5.2: Data sets for test 1

Set	Time (hr)	Start time	End Time
Training	356	02-Feb-95 13:30	17-Feb-95 09:15
Validation	167.25	03-Jan-96 14:45	10-Jan-96 14:15
Test	167.25	10-Jan-96 14:30	17-Jan-96 13:00

The results using these data series are displayed in Fig. 5.19. The efficiency R^2 went from 0.936 in the case of the raingauge data structure, to 0.676 for *matrices*

or structure 3. Among the radar data structures, *average values* had the best performance ($R^2 = 0.828$). It is, nevertheless, lower than that obtained by the model running raingauge data.

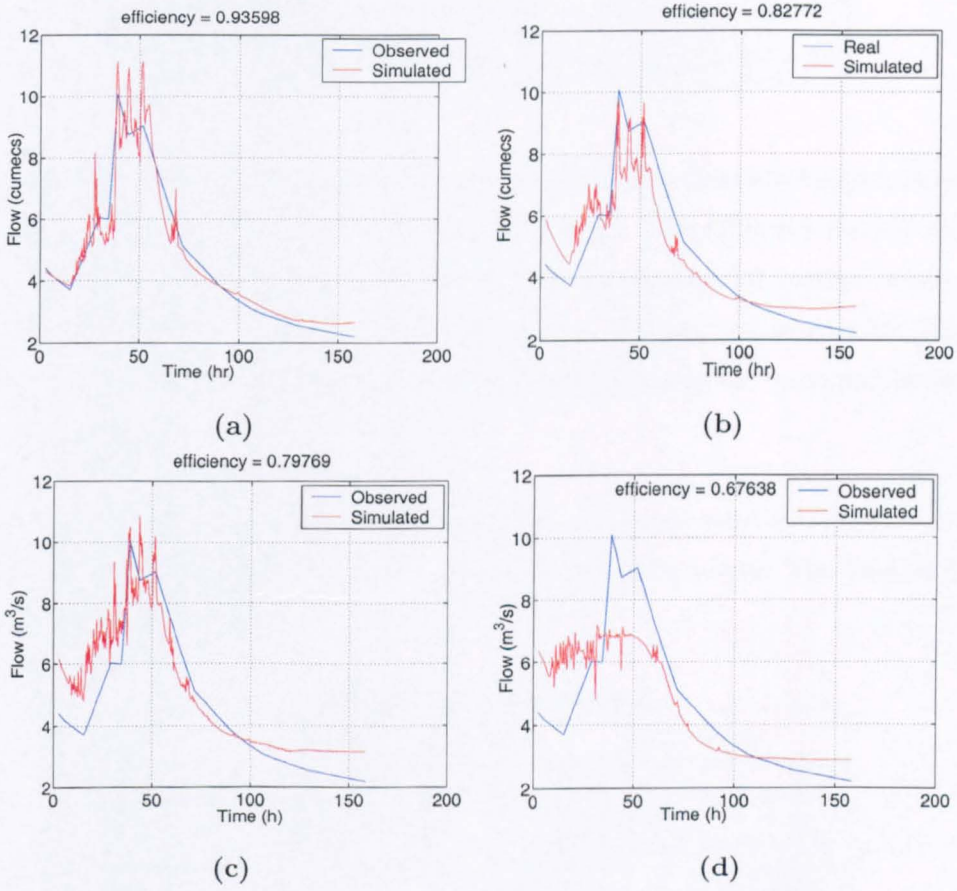


Figure 5.19: ANN model outputs for the period described in Table 5.2. Data were from: (a) raingauge, (b) average radar, (c) matrix and average-values radar, and (d) matrix-series radar.

The second data used to run the ANN model was from winter months too: All from January, but from year 1996 for training the model and from 1998 for testing it. Table 5.3 presents the precise dates and times applied to the model.

The results from these data series are shown in Fig. 5.20. In this case, the

Table 5.3: Data sets for test 2

Set	Time (hr)	Start time	End Time
Training	334.5	03-Jan-96 14:45	17-Jan-96 13:00
Validation	85.75	12-Jan-98 11:30	16-Jan-98 01:15
Test	85.5	16-Jan-98 01:30	19-Jan-98 14:30

efficiency R^2 was lower compared to the previous test. The highest value was likewise from the raingauge data model with an $R^2 = 0.871$. The efficiency for the radar data structures went from 0.693 in the case of the *matrix and average values* or structure 2 to $R^2 = 0.454$ for the *average values*. The performance for the radar data was, as for the previous test, lower than that obtained by the model running raingauge data.

The last data set to be tested came from a summer month: August. This season is characterised by low flow, being the opposite to winter. The time period is presented in Table 5.4.

Table 5.4: Data sets for test 3

Set	Time (hr)	Start time	End Time
Training	300	28-Aug-95 16:15	10-Sep-95 04:00
Validation	85.75	16-Aug-96 18:00	22-Aug-96 01:15
Test	85.5	22-Aug-96 01:30	27-Aug-96 08:45

Running low flow the efficiency decreased dramatically; hence, to get a better assessment of the model performance, the Root Mean Square Error (RMSE) was employed as a second performance indicator. Fig. 5.21 displays the four results from the different data. Raingauge data generated the highest efficiency, $R^2 = 0.195$, and though this value is low compared to the two previous tests, the $RMSE = 0.0248$ demonstrates that the model generated a flow with a small deviation from the observed flow. When testing the model with radar data, in this occasion it

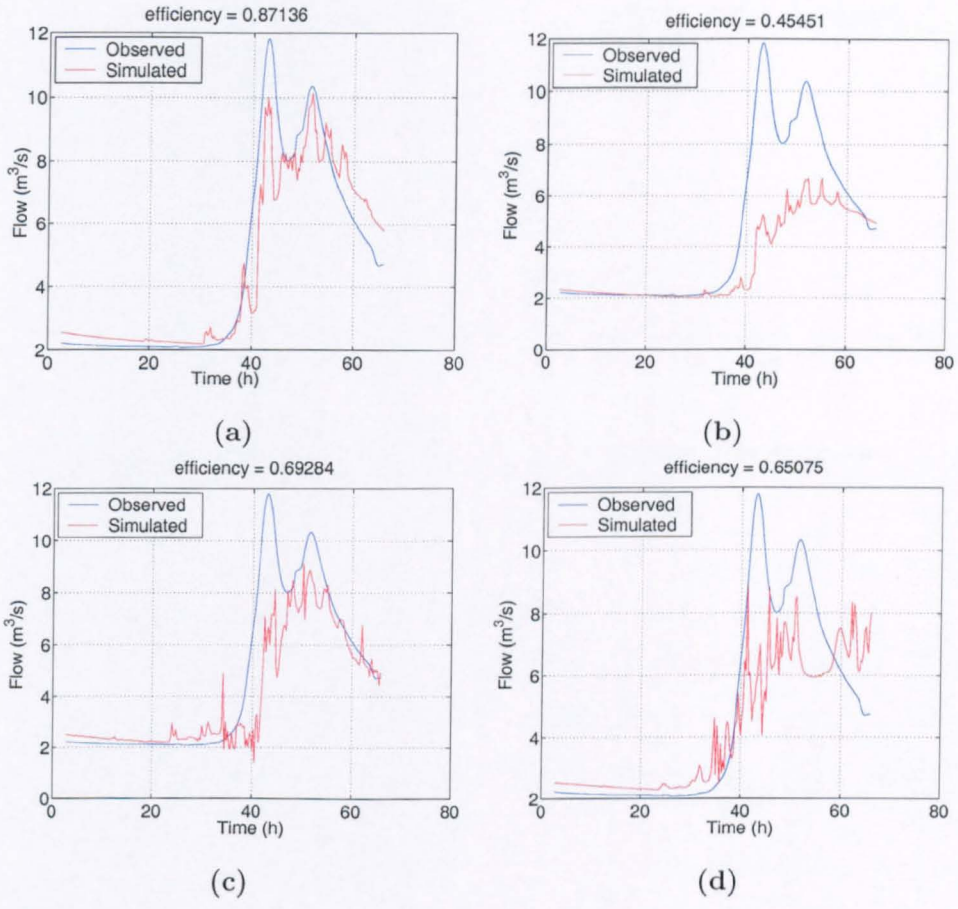


Figure 5.20: ANN model outputs for the period described in Table 5.3. Data were from: (a) raingauge, (b) average radar, (c) matrix and average-values radar, and (d) matrix-series radar.

was the *matrices* the structure with the best efficiency value, $R^2 = -7.603$. The $RMSE = 0.401$ and Fig. 5.21d indicate that, in global terms, the estimation could be considered acceptable (WMO, 1992).

In each one of the three tests presented in this section there was a different radar data structure with the best performance among them. It can be understood as being due to the ability an ANN has to give preference, through adjusting its weights, to those inputs that fit better with the output it is intended to couple

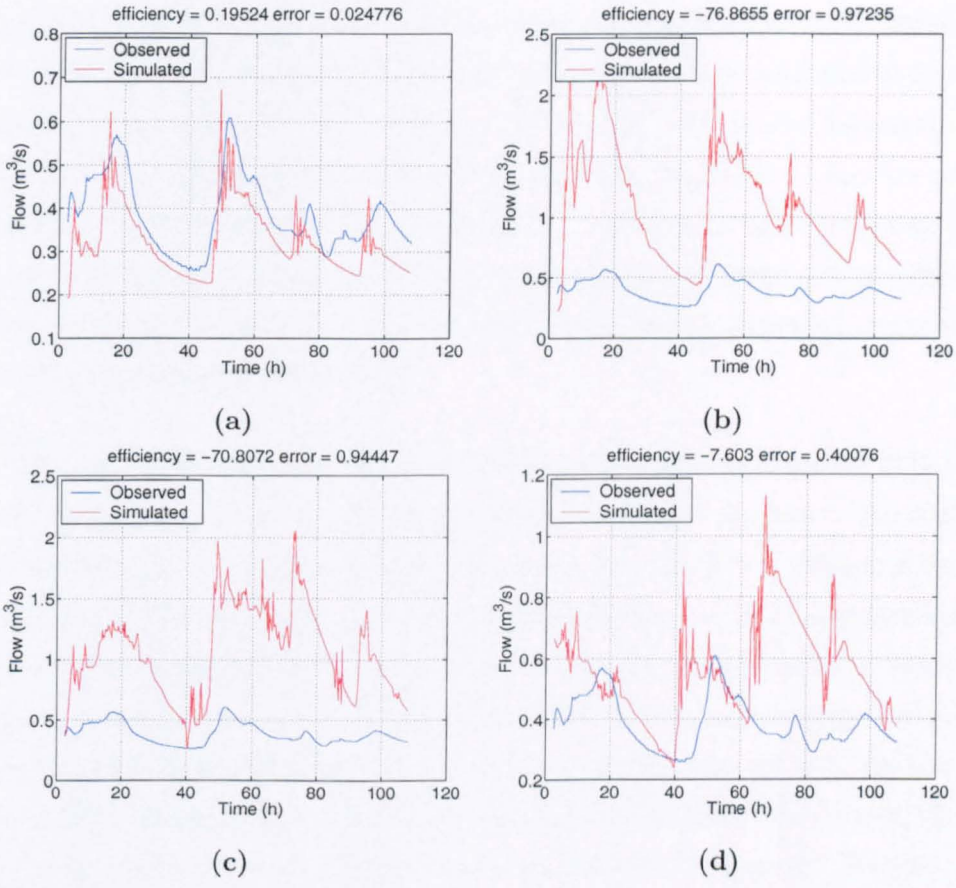


Figure 5.21: ANN model outputs for the period described in Table 5.4. Data were from: (a) raingauge, (b) average radar, (c) matrix and average-values radar, and (d) matrix-series radar.

with. Nevertheless, the performance was always better for the model running with raingauge data.

5.6 Discussion

In remote sensing weather radars are considered active systems since they actively broadcast their own energy and derive information from its reflection and scatter-

ing. They are particularly valuable for assessing relatively large areas or observing inaccessible areas. The setback is the series of problems radars face during operation, as described in Sec. 5.4.2. These problems take a toll on the inaccuracy in rainfall estimation and hence, as showed by the model studied here, the flow estimation. Radar hydrology is an evolving field and a continuous progress is observed in overtaking such problems as described, for example, by Harrison et al. (2000). Any improvement of radar rainfall estimation would allow an extended use on any model using raingauge rainfall data.

The application of radar data in the ANN model, through the three different data structures, allowed an intercomparison on the model response. The model performance varied for each one of the structures and, due to the estimation error associated to the radar data, it is not possible to determine what structure may be employed in the model. If the principle of parsimony is adopted, the *average values* or structure 3, because its simpler structure and implementation, could be chosen; but, it is needed to wait for better radar rainfall estimations to decide on this matter. Another option may be the use of row polar data which are the data generated directly by weather radars before any process is carried out. The task of establishing the relationship between these type of data and runoff data would be left to the ANN model. Meanwhile, rainfall estimations generated by raingauges are the option selected, as it is for most of the flood forecast models.

Chapter 6

River flow modelling with ANN and GA

He who does not know his way to the sea should take a river for his guide.

[Fr., Les rivieres sont des chemins qui marchant et qui portent ou l'on veut aller.]

Blaise Pascal, *Pensees* (VII, 38)

6.1 Introduction

The foundations for modelling water flow in mathematical terms were established by Newton with his Three Laws of Motion and, following him Antoinne Chézy, Claude-Louis Navier, George Gabriel Stokes and Jean Claude Saint-Venant who derived the equations on which the two conceptual models presented here are based.

The three models studied here are classified as mathematical models, but, in order to differentiate the two models based on the laws of physics (HEC-RAS and MIKE 11) from the Artificial Neural Network (ANN) model, the word "conceptual" is employed to refer to the distributed, conceptual and deterministic former models.

In river management and flood forecasting, the application of conceptual models has expanded. In the UK, the Environment Agency has been implementing forecast systems based on them, working mainly with software developed by Delft Hydraulics and the Danish Hydraulic Institute (DHI). Conceptual models are reliable instruments that can produce precise runoff and flow estimates considering the elements affecting them, like dams, pumps, weirs, sluices, etc. In order to implement such systems, a great amount of work is required: a detailed survey of the related region to define river cross sections, the aquatic and riparian vegetation, the structures along the river, the catchment area, etc., digitising this information and building up the model using data series of rainfall, runoff, pump extraction, temperature, etc, from previous years. One such system, implemented by the Environment Agency in the East Anglian region, the Welland and Glen model, is described briefly here. Some sections of this system are used in this thesis to compare the performance of the three models.

These conceptual models are highly sophisticated. Their respective GUIs (Graphic User Interfaces) allow a high level of interaction and present multiple visualisation options, as will be showed later in this chapter. Among the features they offer is the ease with which the models can be calibrated. MIKE 11, for example, can run its autocalibration processes, which is one of the several useful characteristics.

After the introduction of the two conceptual models, their performance and properties are compared with those of the ANN model.

ANNs are considered low pass filters and, even some adaptive filters, like adaptive beamforming, are regarded as a subclass of ANNs (Haykin, 1996). The conceptual

models, based on differential equations, are also low pass filters. In Sec. 6.8 the three models have been examined under several noise levels. The results are presented in the time and frequency domain to ease the comparisons.

A drawback to using ANNs is that they cannot simulate outputs greater than those they were trained with. There have been several proposals to overcome this mathematical limitation; see, for example Maier and Dandy (1998); Imrie et al. (2000); Campolo et al. (1999). The option analysed in this thesis is the use of a different activation input. In Sec. 6.9 the performance of the ANN model is compared, under different amplitudes of input data, for two different activation functions in the output layer.

Most hydrological systems are formed by at least two river branches joining at some point. For that reason, the ANN model has been adapted to receive data flow from two upstream branches to observe the model performance in simulating the flow in the downstream branch.

The last property to be studied is the response of the ANN model to channel and river flow affected by pump extraction. This artificial influence introduces irregular behaviour to any stream flow and, because river modelling is becoming more commonplace globally, it is vital to know if ANN models can cope with this. The study conducted here is intended to give an insight into ANN capabilities on this condition. The ANN input is fed with the extraction rate and its performance is then compared with that obtained without supplying this information, and subsequently evaluating its performance.

6.2 Mike11

MIKE 11 is a professional engineering software package used for the simulation of flows, water quality and sediment transport in estuaries, rivers, irrigation systems,

channels and other water bodies.

6.2.1 Introduction

MIKE 11 is a dynamic, user-friendly one-dimensional modelling tool for the detailed design, management and operation of both simple and complex river and channel systems. It was developed by the Danish Hydrological Institute (DHI) and comprises a fully Windows integrated Graphical User Interface, which conforms to the evolving standards for Windows based software and it is a 32-bit application.

The MIKE 11 system contains six one-dimensional analysis models: (1) hydrodynamic; (2) advection-dispersion; (3) sediment transport; (4) water quality; (5) rainfall-runoff; and (6) flood forecasting.

6.2.2 User interface

The user interface presents the user with eight options:

- File management
- Data entry and editing through several editors
- Simultaneously input/editing of various data types
- Graphical displays and manipulation of input data
- Importing of river network and topography data from ASCII text files
- Coloured horizontal plan graphics for the system data and results
- Animated presentation of results in horizontal, longitudinal and time series plot
- On-line help

6.2.3 Hydraulic analysis

The hydrodynamic module is the nucleus of the MIKE 11 modelling system and forms the basis for most modules. It solves the vertically integrated equations for the conservation of continuity and momentum, i.e. the Saint-Venant equations, using an implicit, finite difference scheme for the computation of unsteady flows in rivers and estuaries.

The module can describe subcritical as well as supercritical flow conditions through a numerical scheme which adapts according to the local flow conditions (in time and space).

Advanced computational modules are included for the description of flow over hydraulic structures, including possibilities to describe structure operation.

The computational scheme is applicable for vertically homogeneous flow conditions extending from steep river flows to tidally influenced estuaries.

6.2.4 Data storage and management

MIKE 11 includes multiple editors each operating on different types of data. Data from these editors must be saved in separate editor files utilising the default MIKE 11 file extensions.

6.2.5 Outputs

MIKE 11 utilises a separate program, MIKE View, for viewing results. This program presents a variety of functions and features for viewing and analysing simulation results produced by the MIKE 11 system. The main presentation features comprise:

- Colour plan plot of the river network
- Longitudinal profiles
- Time series plot
- Animation of water level in cross-sections
- Plot of Q-h relations

6.3 HEC-RAS

The HEC-RAS modelling system was developed as a part of the Hydrologic Engineering Center's "Next Generation" (NexGen) of hydrologic engineering software. The NexGen project encompasses several aspects of hydrologic engineering, including: rainfall-runoff analysis; river hydraulics; reservoir system simulation; flood damage analysis; and real-time river forecasting for reservoir operations.

6.3.1 Introduction

HEC-RAS is an integrated system of software, designed for interactive use in a multitasking, multi-user network environment. The system is comprised of a graphical user interface, separate hydraulic analysis components, data storage and management capabilities, graphics and reporting facilities.

The HEC-RAS system contains two one-dimensional hydraulic analysis components for: (1) steady flow water surface profile computations, and (2) unsteady flow simulation. Both components use a common geometric data representation and common geometric and hydraulic computation routines. In addition to the two hydraulic analysis components, the system contains several hydraulic design features that can be invoked once the basic water surface profiles are computed.

6.3.2 Summary of capabilities

HEC-RAS is designed to perform one-dimensional hydraulic calculations for a full network of natural and constructed channels. The following is a description of the major capabilities of HEC-RAS.

6.3.3 User Interface

The user interface presents six options:

- File management
- Data entry and editing
- Hydraulic analysis
- Tabulation and graphical displays of input and output data
- Reporting facilities
- On-line help

6.3.4 Hydraulic analysis

Steady Flow Water Surface Profiles. This component of the modelling system is intended for calculating water surface profiles for steady gradually varied flow. The system can handle a full network of channels, a dendritic system, or a single river reach. The steady flow component is capable of modelling subcritical, supercritical, and mixed flow regime water surface profiles.

The basic computational procedure is based on the solution of the one-dimensional energy equation. Energy losses are evaluated by friction (Manning's equation) and

contraction/expansion (coefficient multiplied by the change in velocity head). The momentum equation is utilized in situations where the water surface profile is rapidly varied. These situations include mixed flow regime calculations (i.e. hydraulic jumps), hydraulics of bridges, and evaluating profiles at river confluences (stream junctions).

The effects of various obstructions such as bridges, culverts, weirs, and structures in the flood plain may be considered in the computations. The steady flow system is designed for application in flood plain management and flood insurance studies to evaluate floodway inundations. Capabilities are also available for assessing the changes in water surface profiles due to channel alterations and levees.

Special features of the steady flow component include multiple plan analysis, multiple profile computations, and multiple bridge and/or culvert opening analyses.

Unsteady Flow Simulation. This component of the HEC-RAS modelling system is capable of simulating one-dimensional unsteady flow through a full network of open channels. The unsteady flow equation solver is adapted from Dr. Robert L. Barkau's UNET model (HECRAS, 2002a). This unsteady flow component was developed primarily for subcritical flow regime calculations.

The hydraulic calculations for cross-sections, bridges, culverts, and other hydraulic structures that were developed for the steady flow component are incorporated into the unsteady flow module. Additionally, the unsteady flow component has the ability to model storage areas, navigation dams, tunnels, pumping stations, and levee failures.

6.3.5 Data storage and management

Data storage is accomplished through the use of "flat" files (ASCII and binary). User input data are stored in flat files under separate categories of project, plan, geometry,

steady flow, unsteady flow, and sediment data. Output data is predominantly stored in separate binary files.

Data management is accomplished through the user interface. The modeller is required to enter a single filename for the project being developed. Once the project filename is entered, all other files are automatically created and named by the interface as needed. The interface provides for renaming, moving, and deletion of files on a project by project basis.

6.3.6 Outputs

Graphics include X-Y plots of the river system schematic, cross-sections, profiles, rating curves, hydrographs, and many other hydraulic variables. A three-dimensional plot of multiple cross-sections is also provided. Tabular output is available. Users can select from pre-defined tables or develop their own customized tables. All graphical and tabular output can be displayed on the screen, sent directly to a printer (or plotter), or passed through the Windows Clipboard to other software, such as a word-processor or spreadsheet.

Reporting facilities allow for printed output of input data as well as output data. Reports can be customized as to the amount and type of information desired.

6.4 Models evaluation

6.4.1 Comparison of the two conceptual models

Both models use the principles of conservation of mass (continuity), and of conservation of momentum (HECRAS, 2002a; DHI, 2000). These two laws are expressed

mathematically in the form of partial differential equations, the continuity of mass and momentum equations or Saint-Venant [1] equations.

The continuity equation is expressed in both models by Eq. 6.1.

$$\frac{\delta A}{\delta t} + \frac{\delta Q}{\delta x} - q_l = 0 \tag{6.1}$$

where

- Q = flow
- A = flow area
- x = distance measure along the channel
- q_l = the lateral inflow per unit length

The momentum equation is defined slightly differently in each model. Eqs. 6.2 and 6.3 are the MIKE 11 and HEC-RAS versions respectively.

$$\frac{\delta Q}{\delta t} + \frac{\delta \left(\alpha \frac{Q^2}{A}\right)}{\delta x} + gA\frac{\delta h}{\delta x} + \frac{gQ|Q|}{C^2AR} = 0 \tag{6.2}$$

$$\frac{\delta Q}{\delta t} + \frac{\delta QV}{\delta x} + gA\left(\frac{\delta z}{\delta x} + S_f\right) = 0 \tag{6.3}$$

in which

- h = stage above datum

[1] named after Jean Claude Saint-Venant (1797-1886) a French civil engineer.

- C = Chezy resistance coefficient
- R = hydraulic or resistance radius
- α = momentum distribution coefficient
- V = flow velocity
- S_f = friction slope

The terms $\delta h/\delta x$ and $\delta z/\delta x$ are equivalent, hence, the only two discrepant terms that must be equivalent are the friction terms:

$$\frac{gQ|Q|}{C^2AR} \equiv gAS_f \tag{6.4}$$

S_f, the friction slope can be written as:

$$S_f = \frac{Q|Q|n^2}{R^{4/3}A^2} \tag{6.5}$$

where n is the Manning friction coefficient. Substituting Eq. 6.5 into Eq. 6.4, and simplifying, yields the following expression:

$$\frac{1}{C^2R} \equiv \frac{n^2}{R^{4/3}} \tag{6.6}$$

in any hydrology text book the definition of the Chezy resistance coefficient in hydraulic radius and Manning coefficient terms is $C = R^{1/6}/n$, therefore Eq. 6.6 can be expressed as:

$$\frac{n^2}{R^{1/3}R} \equiv \frac{n^2}{R^{4/3}} \tag{6.7}$$

and the discrepancy is now overcome.

The modelling of mixed flow regimes by each model uses a slightly different approach. When the flow passes through the critical depth (Froude number reaches 1) their algorithms, based on calculating derivatives, become unstable. To solve this stability problem they apply a reduction or suppression factor in the momentum equation. For Mike 11 the modified equation is:

$$\frac{\delta Q}{\delta t} + \beta \frac{\delta \left(\alpha \frac{Q^2}{A}\right)}{\delta x} + gA\frac{\delta h}{\delta x} + \text{friction} = 0 \tag{6.8}$$

where β, the factor for suppression of convective momentum term has two formulations, the following one is implemented for most flood routing studies where the Froude number is relatively small and the space step is greater that the water depth:

$$\beta = \begin{cases} 1 - F^2 & \text{for} \quad F \leq 1 \\ 0 & \text{for} \quad F > 1 \end{cases} \tag{6.9}$$

and, for high Froude number and small space steps, the alternative formulation is:

$$\beta = \begin{cases} 1 & \text{for} \quad F \leq a \\ 1/(F + 1 - a)^b & \text{for} \quad F > a \end{cases} \tag{6.10}$$

where

- F = Froude number
- a = parameter with default value = 1
- b = parameter with default value = 2

For HEC-RAS the modified momentum equation is:

$$\sigma\left(\frac{\delta Q}{\delta t}+\frac{\delta QV}{\delta x}\right)+gA\left(\frac{\delta z}{\delta x}+\text{friction}\right)=0 \tag{6.11}$$

and

$$\sigma=\begin{cases} F_T-F_r^m & \text{for} \quad F_r \leq F_T; m \geq 1 \\ 0 & \text{for} \quad F_r > F_T \end{cases} \tag{6.12}$$

where

- σ = reducing factor
- F_T = Froude number threshold at which factor is set to zero. This value should range from 1.0 to 2.0 (default is 1.0)
- F_r = Froude number
- m = Exponent of equation, which changes the shape of the curve. This exponent can range from 1 to 128 (default value is 10)

Eqs. 6.8 and 6.11 show that MIKE 11 applies its suppression factor only to the convective term and HEC-RAS does it to the convective and local acceleration terms. The behaviour of β and σ are plotted in relation to different Froude number values in Figs. 6.1 and 6.2; the former representing MIKE 11 equations and the latter HEC-RAS equation, which is plotted for two Froude number thresholds, 1 and 2. The other parameters involved in these equations retain their default values.

There are more differences in the hydraulic analysis between both models, but describing them goes beyond the scope of this work. Before presenting a real modelling system, an idealised example is presented to observe the models' behaviour under controlled and precise conditions.

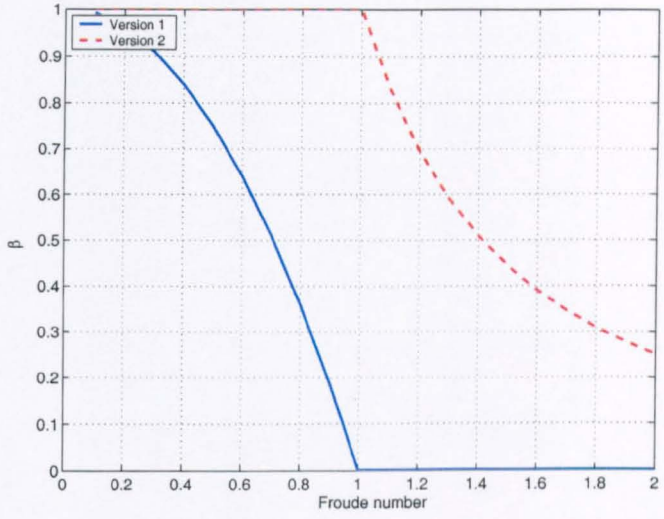


Figure 6.1: Plot showing both β factors implemented in MIKE 11.

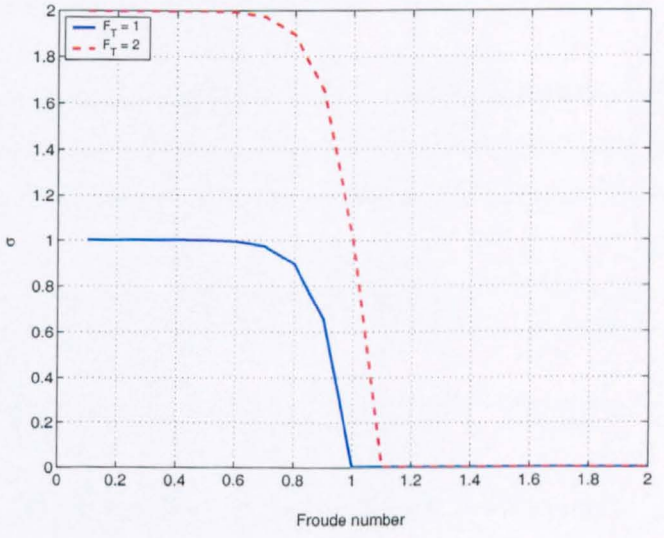


Figure 6.2: Plot showing σ factor implemented in HEC-RAS.

Table 6.1: Channel features in both models

Feature	Value
Channel extension	48 km
Channel slope	1/1000
Number of cross sections	49
Cross sections shape	square
Cross sections hight	5 m
Cross sections width	5 m
Manning value n	0.03
Computation interval	15 minutes

6.4.2 An ideal study case

A channel with ideal conditions is built up in both programs to observe the implementation of the Saint-Venant equations. This channel has the characteristics presented in Table 6.1.

The input hydrograph applied to both models is presented in Fig. 6.3. This simple hydrograph was intended to observe the differences, if any, of the models' outputs. The general output obtained by HEC-RAS is displayed in Fig. 6.4.

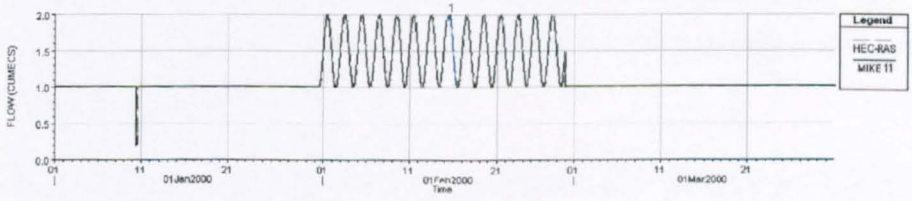


Figure 6.3: Input hydrograph applied to both models.

Two detailed views of the flow comparing outputs from MIKE 11 and HEC-RAS are presented in Figs. 6.5 and 6.6. The hydrograph of Fig. 6.6 revealed

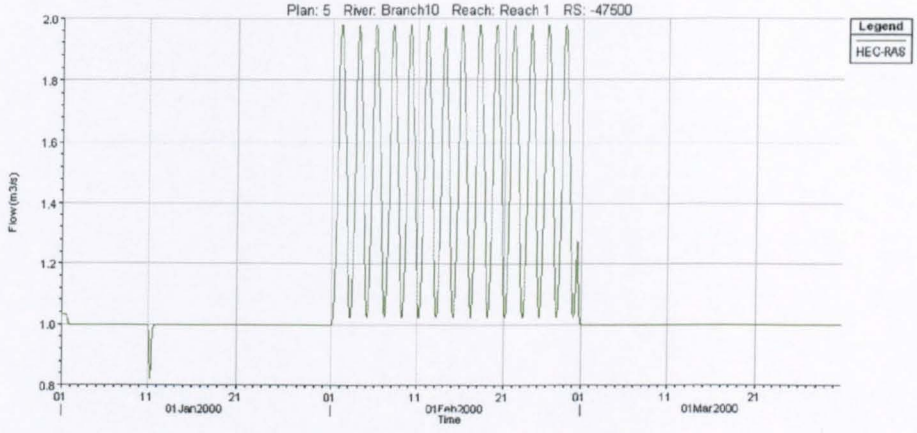


Figure 6.4: Output hydrograph by HEC-RAS.

both attenuation and delay of the flood wave generated by the HEC-RAS model in relation to the flood wave generated by MIKE 11. To make both peak flows coincide with each other, it was necessary to adjust the Manning coefficient and θ in HEC-RAS. θ is a weighting applied to the finite difference approximations when solving the unsteady flow equations. In HECRAS (2002b) it is recommended a practical limit for this parameter from 0.6 to 1.0. The modified values in HEC-RAS were $\theta = 0.6$ and the Manning factor $n = 0.021$ (the Manning factor is 70% the value used in MIKE 11). The resulting hydrograph is plotted in Fig. 6.7

After the parameter modifications the two peak flows coincided in time and height but the rising and receding limbs of the HEC-RAS hydrograph had a delay of up to 3:20 hours at their base.

The last analysis conducted on the ideal model was the generation of critical and supercritical flow. To reach these conditions the parameter modified was the channel's slope. In order to estimate the required slope to reach critical conditions, the following two equations, taken from any hydraulic book, for example the book by Chadwick and Morfett (1993), have been combined: Eq. 6.13 calculates discharge for uniform flow and Eq. 6.14 gives the conditions for critical flow:

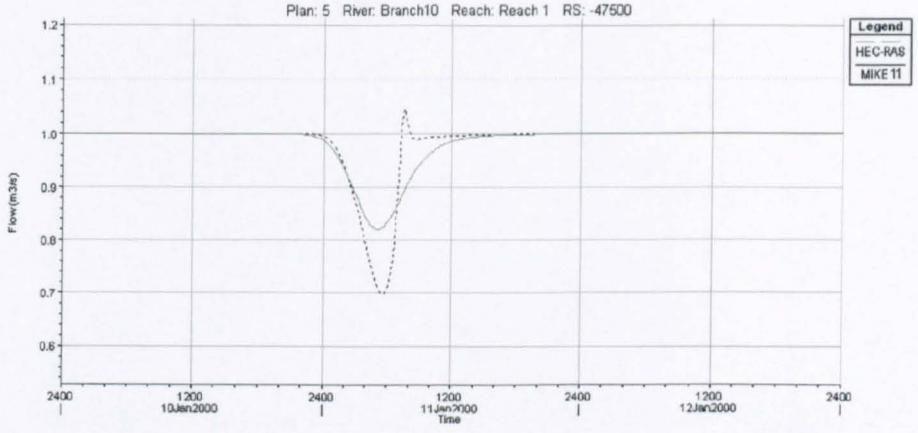


Figure 6.5: Detailed output by HEC-RAS and MIKE 11.

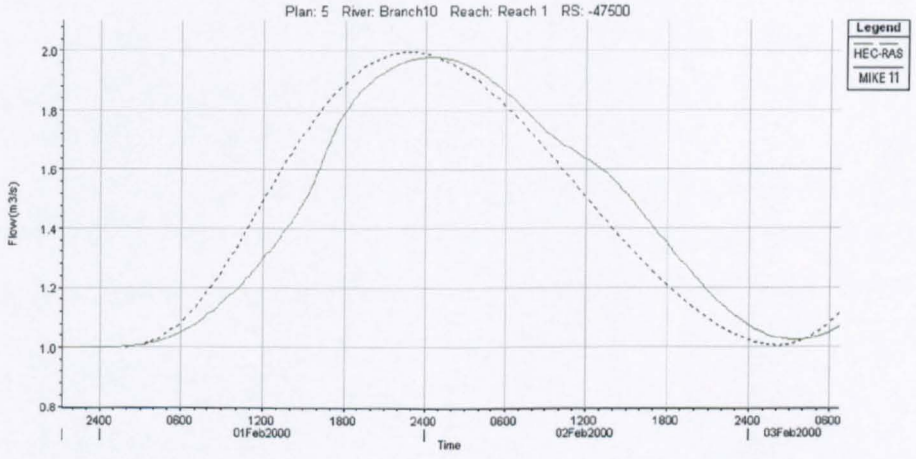


Figure 6.6: Detailed output by HEC-RAS and MIKE 11.

$$Q = \frac{1}{n}\frac{A^{5/3}}{P^{2/3}}S_0^{1/2} \tag{6.13}$$

and

$$\frac{Q^2 B}{gA^3} = 1 \tag{6.14}$$

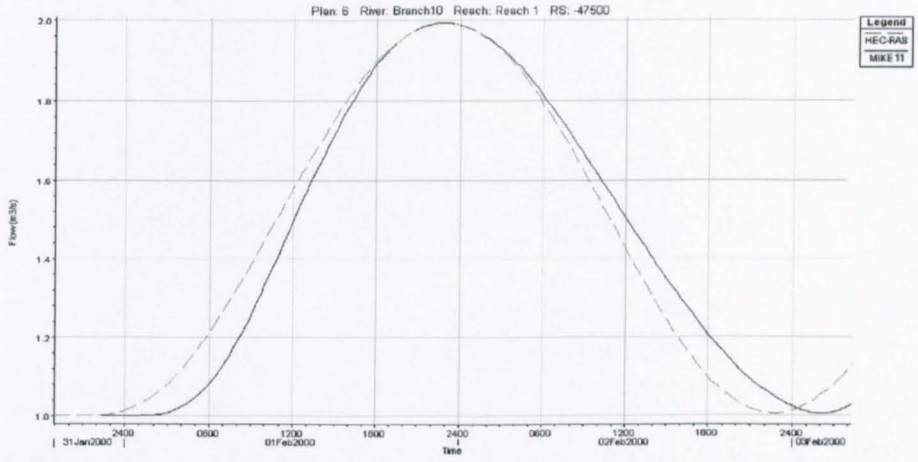


Figure 6.7: Detailed output by both models after the HEC-RAS parameters have been modified.

where

- Q = discharge
- A = cross-sectional area of flow
- P = wetted perimeter
- S_0 = channel slope
- B = surface width
- n = Manning factor
- g = gravity

For a rectangular channel of width b, $B = b$, $P = b + 2y$ where y is the depth from the free surface to the bottom of the channel.

Combining 6.13 and 6.14 to eliminate A,

$$S_c^{1/2} = \frac{n g^{5/9} P^{2/3}}{Q^{1/9} B^{5/9}} \tag{6.15}$$

if it is assumed a wide rectangular channel with $b >> y$, $P \simeq b$ and, making the substitutions and simplifying, the critical slope S_c was obtained by:

$$S_c = \frac{n^2 g^{10/9} b^{2/9}}{Q^{2/9}} \tag{6.16}$$

Applying the conditions of the model to Eq. 6.16: $n = 0.03$, $b = 5m$, and using a discharge $Q = 1m^3/s$, the critical slope would be $S_c = 0.01627$. For the highest discharge value applied to the model, that is $Q = 2m^3/s$, $S_c = 0.01394$. The change in slope was done by changing the datum to two cross sections (at 5000m and 4000m)from 1 to 15 m. MIKE 11 made the calculations without problem but HEC-RAS became unstable, even reducing the time step to 30 sec. A further change has been done to the immediate cross sections upstream and downstream of the previously modified cross sections (a datum modification to 3m for both cross sections: at 6000m and at 3000m). The final profile used by both models is displayed in Fig. 6.8 and a detail graph including the water surface profile generated by HEC-RAS is shown in Fig. 6.9

The Froude numbers generated after applying these new conditions were different between both models at the locations under study. Under steady conditions ($Q = 1m/s$) and setting the exponent $m = 2$ from Eq. 6.12 to equalise Eq. 6.9, Eqs. 6.8 and 6.11 became identical because the local acceleration term $\delta Q/\delta t$ was cancelled. Nevertheless, as it is shown in Figs. 6.10 and 6.11 the Froude number at the 5000m point, for MIKE 11 was $F_r = 1.009$ and, for HEC-RAS $F_r = 2.34$.

When the flow was increased to $Q = 2m^3/s$ the Froude number changed to $F_r = 1.66$ for HEC-RAS and $F_r = 1.082$ for MIKE 11. Since an increase in flow, keeping all the other conditions intact, would generate an increase in the Froude

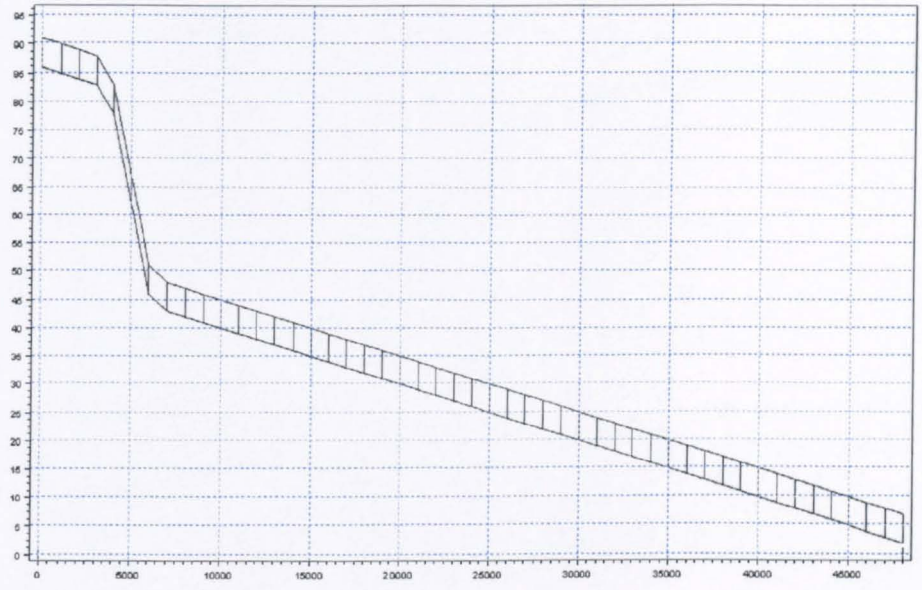


Figure 6.8: Profile of ideal channel including the increased slope.

number, HEC-RAS produced the opposite result. MIKE 11 results were closer to expected values. The manuals of both models do not describe in detail the operations under critical and supercritical conditions, hence, it is not possible to analyse the mathematical implementations developed for the models.

6.4.3 Comparison comments

This comparison does not consider three analysis models present in HEC-RAS: advection-dispersion, sediment transport and water quality. It is focused only on the flow routing model.

The two models utilise the Saint-Venant equations but, their implementation is slightly different and, hence the results, under identical conditions, are not the same. Likewise, as it is described in the previous section, under critical and supercritical conditions the results differ significantly from each other.

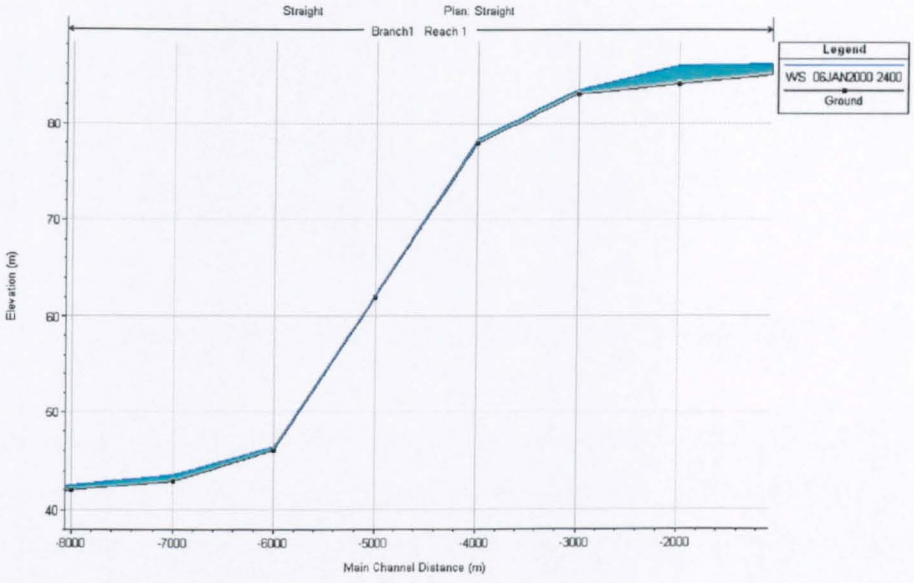


Figure 6.9: Detailed profile of ideal channel between cross sections at 2000m and 8000m.

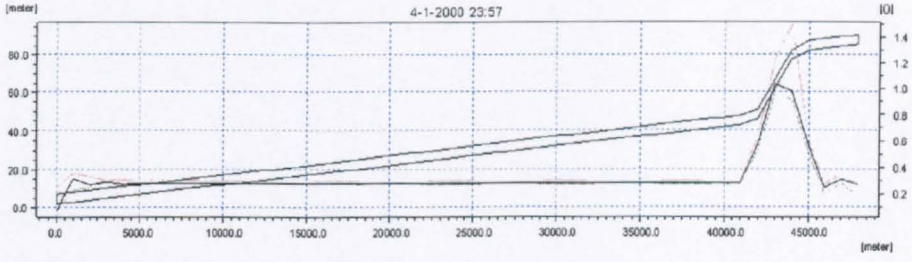


Figure 6.10: Froude number profile generated by MIKE 11.

MIKE 11 produced the most reliable results, though manipulating parameters and flow conditions, within subcritical flow, HEC-RAS improved its results. MIKE 11 includes too, three models for rainfall-runoff modelling: NAM, UHM, and SMAP. The U.S. Army Corps of Engineers has a separate program for this purpose: HEC-HMS.

The displays in both models have almost identical features; a nice and useful option is the 3D view of the whole model (or any section) in HEC-RAS. In every

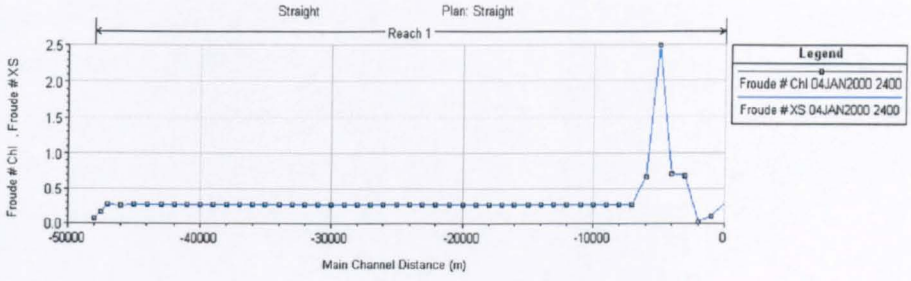


Figure 6.11: Froude number profile generated by HEC-RAS.

graph, selecting a point with the mouse, HEC-RAS shows the values for that specific point. In MIKE VIEW for displaying a cross section the user selects, from the river plan, the desired cross section but, once it is displayed, the image does not show what cross section it is.

For introducing data series and manipulating them, both models have facilities to do so. Importing model data from MIKE 11 to HEC-RAS was not a straightforward step, but was eased by the facilities provided by HEC-RAS. The Welland and Glen Model described in the next section is formed by 1915 points and it was not possible to import it all to HEC-RAS. In the model's documentation there is no indication about the maximum number of points it can accept. Selecting just a section of the Welland and Glen model was the preferred option not just for overcoming this restriction but for computational limitations too.

HEC-RAS limits the postprocessor profiles to 2000; hence, the only available profiles after 2000 steps are the flow and stage water surface profiles generated by the Unsteady Flow Simulation. For the 15-minute time step used in all the simulations, the postprocessor profiles are generated for just 20 days and 20 hours. MIKE 11 does not have this restriction.

The great advantage of HEC-RAS is that it can be downloaded free of charge. If support is required, it can obtained through several vendors (BOSS International

charges $95 for, according to its own information, a lifetime technical support). On the other hand, MIKE 11 is a commercial package, with full support by DHI representatives, with a cost of approximately £2000. Notwithstanding, for big projects like the Welland and Glen model, where many people are involved and several years are required to implement it, the software cost is secondary.

6.5 Welland and Glen Model

The Welland and Glen Model was commissioned by The Environment Agency through WS Atkins Consultants and the Danish Hydraulic Institute. It serves as a Flood Forecasting System for the area covered by the catchment. Following is a description of the catchment.

The Welland and Glen catchment is located in the Anglian Region. The River Welland rises in Leicestershire and flows eastwards up to its outfall into the Wash Estuary. The River Glen rises in the Lincolnshire Ridge east of Grantham, flowing through several villages before it meets the River Welland at Surfleet Seas End.

Both rivers and their tributaries drain an area of approximately 1150 square km. There are several population centres within the catchment, the most significant being Oakham, Stamford, Spalding, Market Harborough and the northern fringes of Peterborough.

The geology and landscape of the area vary considerably as one moves from west to east. To the west of Stamford the Welland catchment is dominated by undulating low lying hills with the river's major tributaries cutting valleys through overlying boulder-clay into Lias clays. The upper Glen catchment is also slightly hilly and overlies a complex geology of limestone, clays and minor gravel aquifers. East of Stamford the catchments of the two river system fall into the very different flat landscape of the Fens as the Lincolnshire limestone dips.

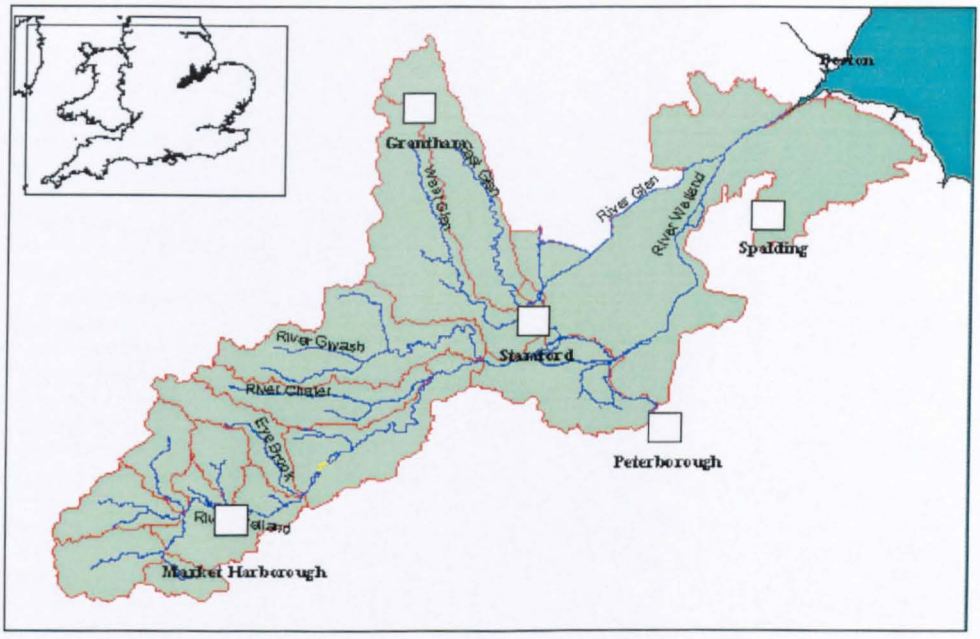


Figure 6.12: The Welland and Glen catchment.

The Welland is tidal downstream of Marsh Road Sluice in Spalding. The tidal limit of the Glen is at Surfleet Sluice, the confluence of the two rivers. Downstream of Surfleet the river takes the name of the Welland and discharges in the Wash at Fosdyke Bridge.

In order to perform the analysis of the two conceptual models, because it was not possible to set up the whole model in the HEC-RAS program, a section comprising the Upper East Glen has been selected. The next section describes the technical details of this catchment and presents the simulations generated by both programs.

6.5.1 Upper East Glen

The section of the river Glen modelled in both software programs stretches over 8723 m., between Irnham and Edenham in Lincolnshire, see Fig. 4.11. Runoff from

Table 6.2: River Glen's characteristics

Feature	Value	Comment
River extension	8723 m	
Cross sections	30	for HEC-RAS the number is 48
Weirs	2	at 15, and 5029 (m)
Culverts	6	at -10, 675, 1725, 5029,6175, and 6952 (m)
Runoff links	2	models 8 and 8a
Runoff area for model 8	$65.98km^2$	added at -15m
Runoff area for model 8a	$35.9km^2$	added from -15 to 8723 (m)
Bed resistance	0.033	excluding structures
Output time interval	15 min	

the catchment is divided in two subcatchments: the region upstream of Irnham is simulated by a NAM model numbered 8 and is added to the model at the chainage -15[2]. Between Irnham and Edenham is represented by the NAM model 8a and the flow is added along all the chainages included in the model. The characteristics of the model are presented in Table 6.2

The implementation and calibration of the model was carried out by The Environment Agency and its contractors on MIKE 11. The simulation plots showing the river profile and its hydrograph between January and April 1998 are displayed in Figs. 6.13 and 6.14.

The model was transferred to HEC-RAS leaving almost all the parameters identical. A change had to be done to an extraction flow implemented in the MIKE 11 model for stability. This outflow was applied in chainage 6947 with a constant value

[2]In MIKE 11 locations along the river are referred to as 'chainages' increasing downstream with positive values. In HEC-RAS they are referred to as 'river stations', increasing downstream but with negative values. The figures indicate meters

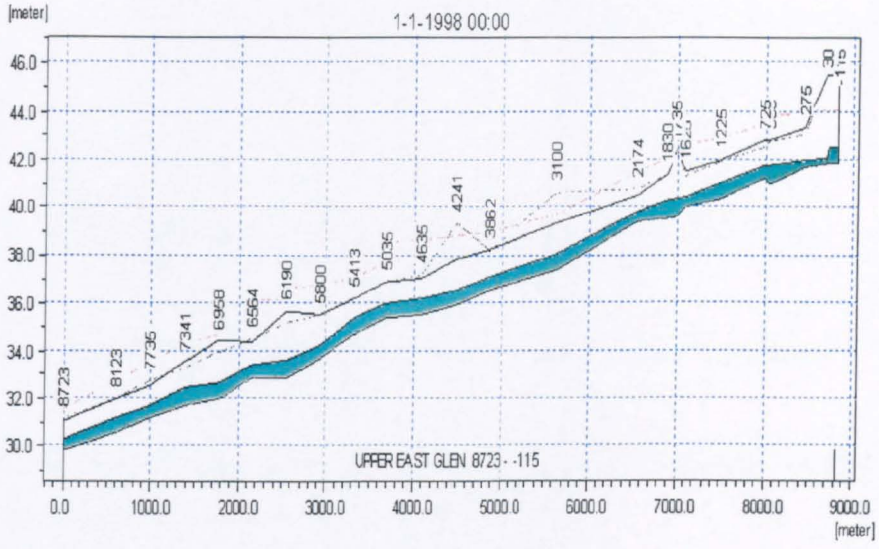


Figure 6.13: Upper East Glen profile generated by MIKE 11.

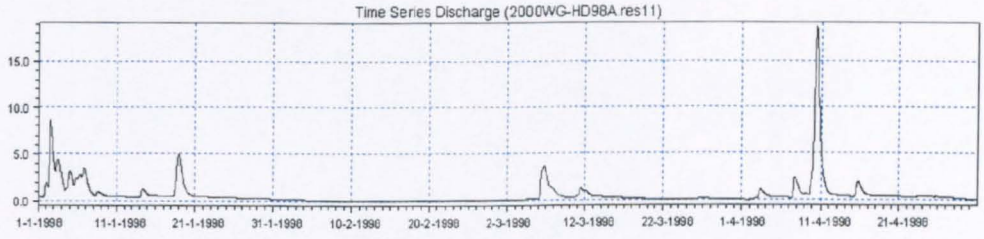


Figure 6.14: Upper East Glen flow from 01/01/1998 to 30/04/1998.

of 0.5 m^3/s. In HEC-RAS this value was reduced to 0.05 m^3/s in order to guarantee the stability of the model.

Fig. 6.15 displays the flow generated by HEC-RAS for the period January-April 1998 at river station 7929. To the model output has been added the output generated by MIKE 11 to compare both results.

A more detailed view is displayed in Fig. 6.16. In this graph the reduced extraction in the HEC-RAS model is clearly visible with a greater flow (approx.

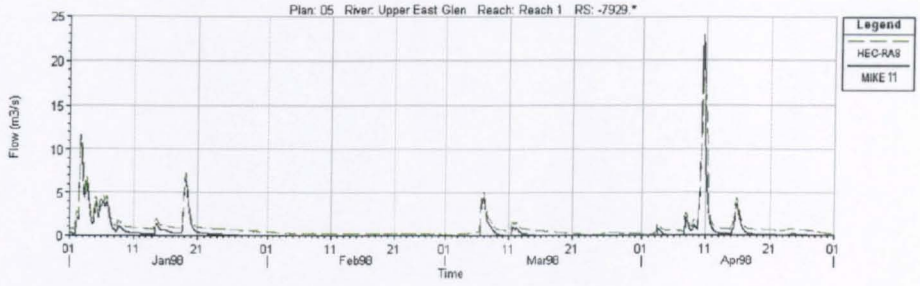


Figure 6.15: Upper East Glen flow from 01/01/1998 to 30/04/1998.

0.4 $m^3/2$). Despite this increase in flow the peak of the hydrograph on the 2nd of January was lower than its counterpart by MIKE 11.

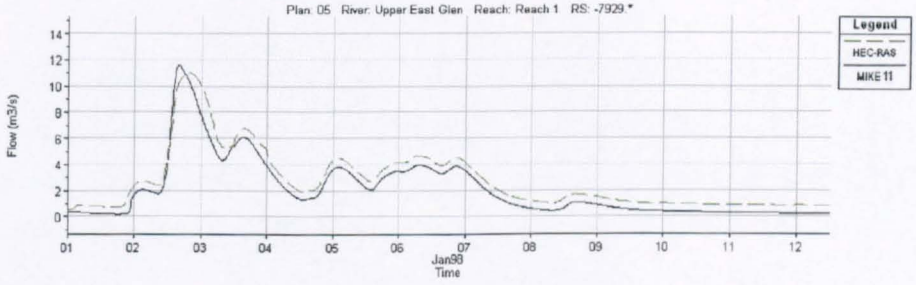


Figure 6.16: Upper East Glen flow for January 1998.

Fig. 6.17 shows the monthly flow for April 1998. For the whole period but the 10th of April, the HEC-RAS result showed a continuous, greater flow. The hydrograph for the 10th, the biggest peak of the entire period, showed a marked discrepancy between the two models. The peak flow for the hydrograph by MIKE 11 had a value of $Q = 22.97m^3/s$ and, for the hydrograph by HEC-RAS it was $Q = 17.95m^3/s$. The receding limb between both hydrographs had a maximum time difference of 8.5 hours.

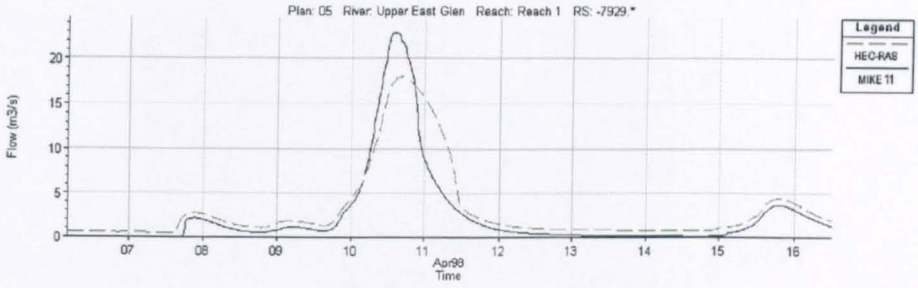


Figure 6.17: Upper East Glen flow for April 1998.

6.6 ANN model simulations using conceptual models outputs

The ANN model was employed to simulate the discharge between two extreme sections of the river. The points selected were at the chainage 152.5 and the chainage 7929. The Genetic Algorithm described in Chapter 4 generated the parameters for the model presented in Table 6.3

Table 6.3: ANN parameters

Parameter	Description	Value
N_l	number of layers	3
N_n	number of neurons on each layer	6 - 16 - 1
A_l	average-previous flow values	1
N_a	number of average data	60

The lead time between points (chainages 152.5 and 7929) was two hours, it means 8 time steps of 15-minute each. The training, testing and validation periods are presented in Table 6.4:

The period selected for training the ANN model covered the whole range of flow values for this series and it allowed the ANN model to simulate low and high values

Table 6.4: Training, testing and validation periods (1998)

Set	Time (hr)	Start time	End Time
Training	400	01-Apr 14:15	18-Apr 06:00
Validation	250	01-Jan 00:00	11-Jan 09:45
Test	250	11-Jan 10:00	21-Jan 19:45

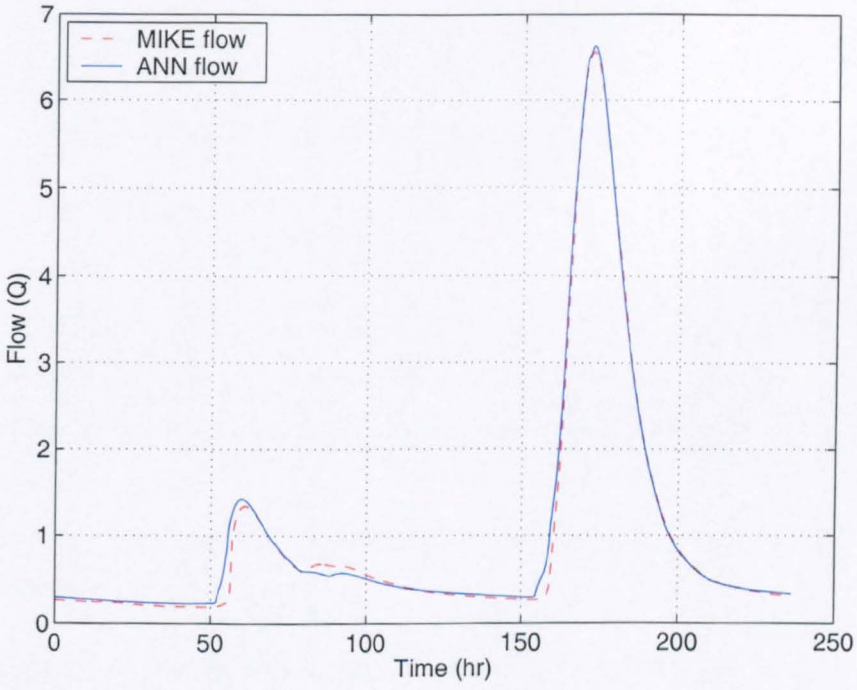


Figure 6.18: Upper East Glen discharge simulated by MIKE 11 and the ANN model.

on the validation period. Figure 6.18 displays the flow generated by MIKE 11 and the ANN model during the validation period. The high precision can be noted on simulating the peak of the highest hydrograph by the ANN model. The efficiency value for this period was $R^2 = 0.9936$.

The results generated by HEC-RAS were, likewise, simulated by the ANN model. The same conditions have been employed in simulating the MIKE 11 results. The

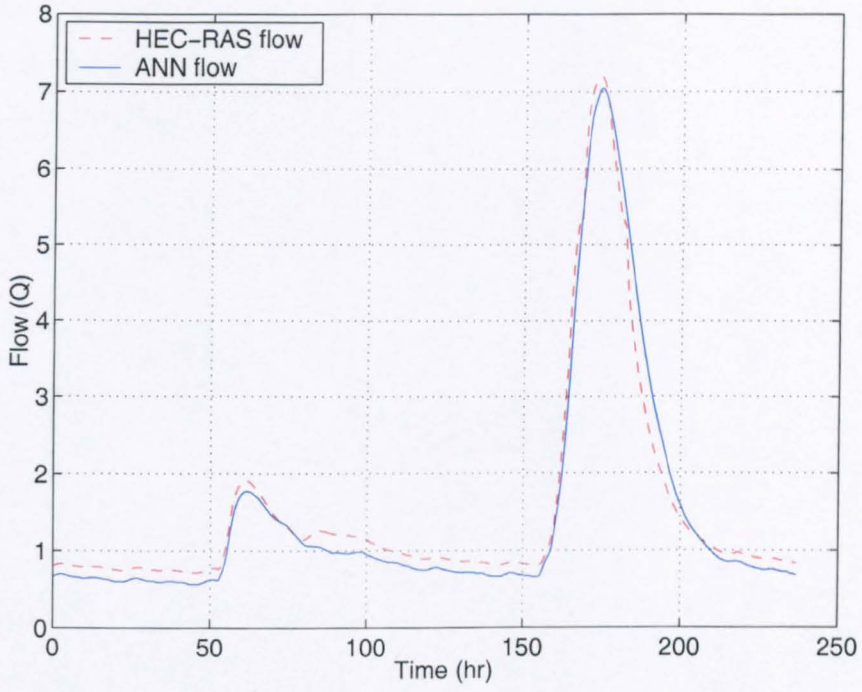


Figure 6.19: Upper East Glen discharge simulated by HEC-RAS and the ANN model.

efficiency dropped from 0.9936 with MIKE 11 output to 0.9765 with HEC-RAS. The results are plotted in Fig. 6.19

With this test the ANN model was run along one single branch, using one input and one output. The following section extends the test to the case of two branches joining into one.

6.7 ANN model simulation of a river junction

A river or channel system is formed almost always by a network of branches which, at some point, has junctions. In this section, the ANN model was tested on simulating the flow of a river coming upstream from two separate branches. For this purpose, data from the Welland and Glen model run by Mike 11 has been used.

The river branches under study were the section of the river Welland classified in the model as A15S and the flood relief channel Maxey Cut. Fig. 6.20 shows the location of the points used in the simulation of the junction.

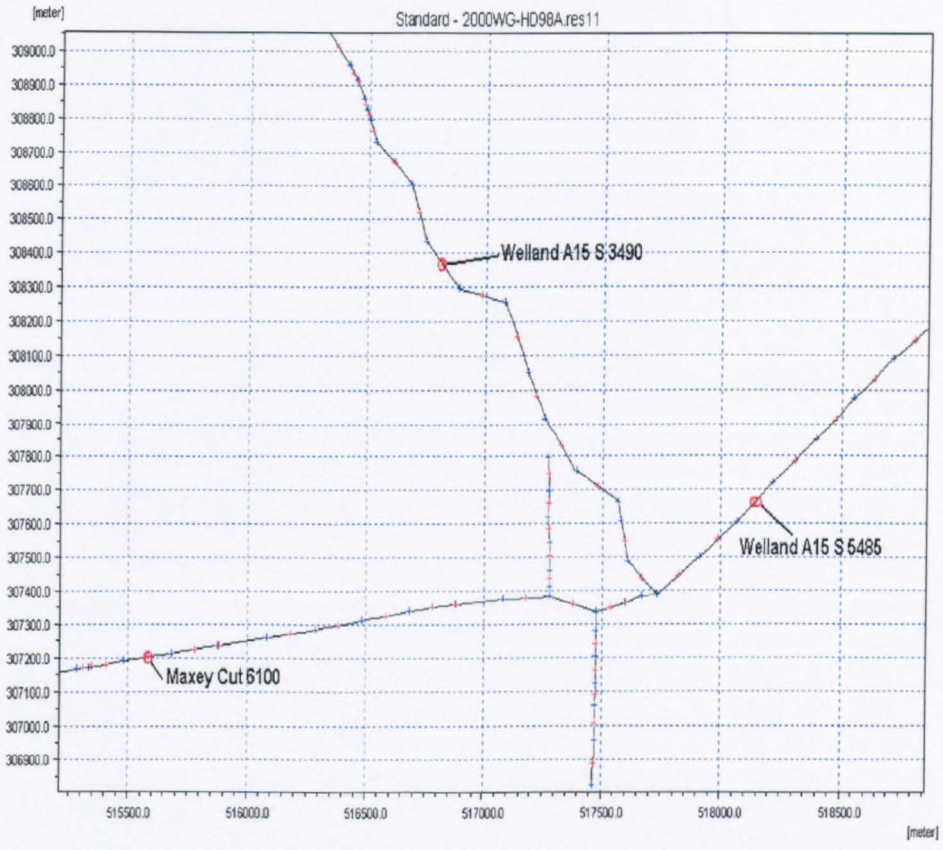


Figure 6.20: Schematic section of the Welland and Glen catchment showing the points used in the simulation of the junction.

Table 6.5 presents the main characteristics of the river Welland and the channel Maxey Cut.

The time delays were taken as the travel times for the highest flow peaks from the chainages before the junction to the chainage downstream, all where the data were taken from.

Table 6.5: Description of River Welland and channel Maxey cut

Feature	Description
Maxey Cut extension	8200 m
spans out	from Tallington to Welland A15S at chainage 4990 m
data taken from	chainage 6100
Time delay to forecast point	2 hr. (4 time steps)
Welland A15S extension	23880 m
spans out	from Marsh Rd Sluice to West Deeping Cut
data taken from	chainage 3490 (before the junction)
	chainage 5485 (after the junction)
Time delay to forecast point	2.25 hr. (5 time steps)
Output time interval	15 min

The ANN model was based on the previous model used to simulate the river Glen. The number of inputs was double to enter the information from the two upstream flows. The number of hidden layers and the number of average data were left to be determined by the genetic algorithm. The ANN model architecture is specified in Table 6.6.

Table 6.6: Parameters of the ANN model

Parameter	Description	Value
N_l	number of layers	4
N_n	number of neurons on each layer	12 - 12 - 12 - 1
A_l	previous average-flow inputs	1 from each river
N_a	number of average data	80 and 60

The data set was from the first months of 1998. The specific dates and number of hours are detailed in Table 6.7.

Fig. 6.21 shows three plots: a) is the flow from river Welland A15S used as input

Table 6.7: Training, testing and validation periods (1998)

Set	Time (hr)	Start time	End Time
Training	750	01-Jan 00:00	01-Feb 05:45
Validation	750	01-Feb 06:00	04-Mar 11:45
Test	750	04-Mar 12:00	04-Apr 17:45

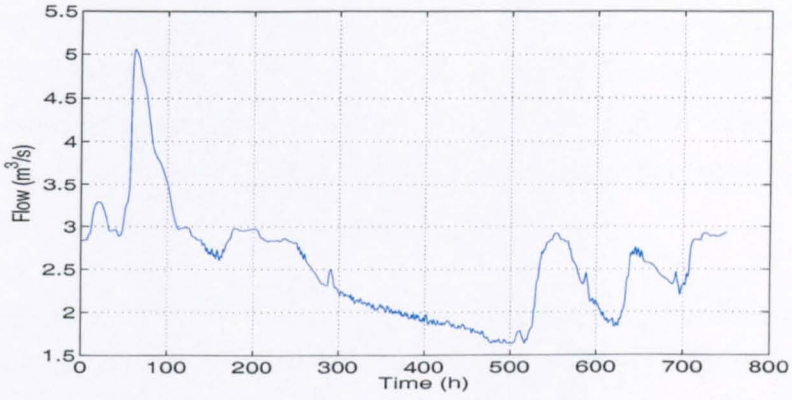
to the model; b) is the flow of the channel Maxey Cut used likewise as input to the model; c) is the estimated flow generated by the model.

The efficiency was, like for the river Glen, high with $R^2 = 0.99141$. From the plots it is clear that the flow downstream of the river Welland was influenced mainly by the flow coming from the channel Maxey Cut but the ANN model ability to sum up or inhibit inputs allowed it to choose the most influential values.

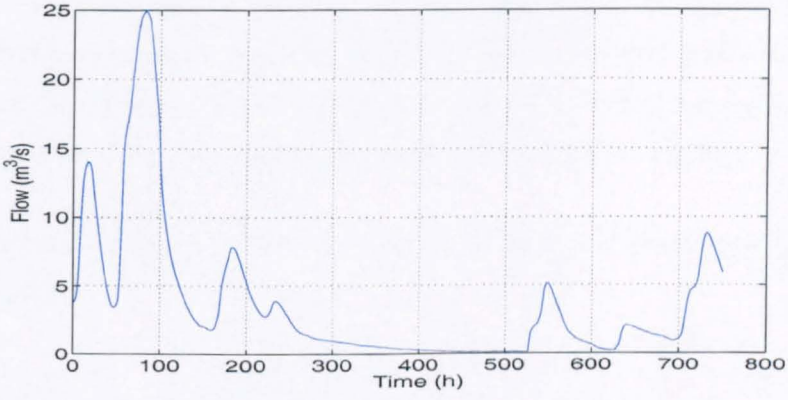
The two previous tests were performed under normal conditions generated by the Welland and Glen model. The following sections describe the analysis and tests performed on the ANN model applying artificial conditions to study the model under extreme circumstances. In the first case White Gaussian Noise (WGN) was applied to the three models in order to compare their response and assess adequately the ANN model.

6.8 Response to noise by the three models

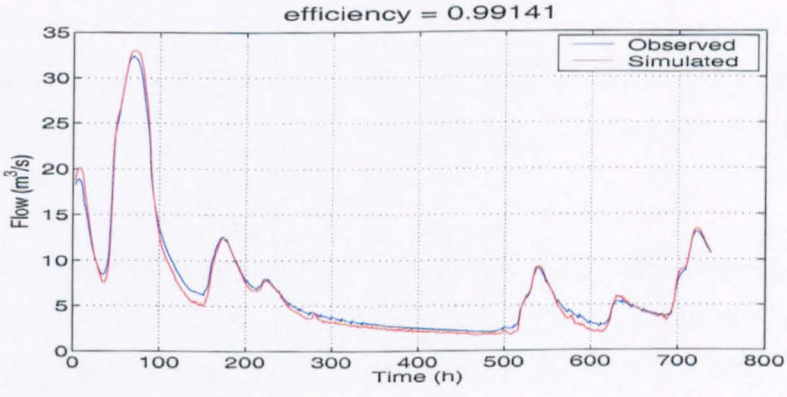
Every electrical and/or electronic system has inherent noise sources. This noise is caused by the chaotic motion of electrons in the components of the system, and is sometimes referred to as Thermal noise. The first person to measure Thermal noise in conductors was J. B. Johnson in 1928 while working at Bell Labs. His experiments showed that the power spectral density of Thermal noise was over a



(a)



(b)



(c)

Figure 6.21: ANN model outputs for the test period described in Table 6.6. (a) input flow from Welland A15S at chainage 3490, (b) input flow from Maxey Cut at chainage 6100, (c) output flow from Welland A15S at chainage 5485.

large range of frequencies (as high as 10^{13} Hz). He explained his findings to H. Nyquist who also worked at Bell Labs. Nyquist derived an expression by means of thermodynamics and statistical mechanics which fitted Johnson's measurements. The work of these two men has led to Thermal Noise being called Johnson Noise by experimental physicists and Nyquist Noise by theoreticians.

Though digitised systems have reduced the effect of Thermal Noise it is still present and in the case of hydrological models designed to operate on a real-time basis, remote sensing noise, caused by data errors and/or data holes, is another important source of noise. In a smaller proportion, the quantisation process brings an additional associated level of noise. Hence, in order to test the models under noisy conditions, the idealised version of Thermal noise, White Gaussian Noise (WGN), is applied to the three models and the results analysed and compared.

Following is the mathematical description of WGN and afterwards its application to the models.

6.8.1 White Gaussian noise

White Gaussian noise takes its name from white light which is a combination of all colours and described by a probability distribution that follows a gaussian bell shaped curve.

It is a stochastic process with random variables temporally decorrelated; that is, at different times they are mutually independent (Deco and Schürmann, 2001). This property is mathematically described using its Probability Density Function (PDF) as below:

$$p(x_{t_1}, x_{t_2}) = p(x_{t_1})p(x_{t_2}) \tag{6.17}$$

and with equal PDFs defined as follow:

$$p(x_{t_1}) = p(x_{t_2}) \tag{6.18}$$

This noise is Gaussian with mean $m = 0$ and variance σ^2. The PDF $p(x_t)$ is a normal distribution compactly written as $N(0, \sigma^2)$, and described by the following equation:

$$p(x_t) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) e^{\frac{-x_t^2}{2\sigma^2}} \tag{6.19}$$

The mean value m is expressed by:

$$\langle x_t \rangle = m = \int dx_t \, x_t \, p(x_t) = 0 \tag{6.20}$$

and the autocorrelation satisfies the equation:

$$\langle x_t \, x_{t'} \rangle = \int \int dx_t \, dx_{t'} \, x_t \, x_{t'} \, p(x_t) \, p(x_{t'}) = \sigma^2 \delta(t - t') \tag{6.21}$$

WGN comprises, ideally, all frequencies in the range $[-\pi, \pi]$ and the power spectral density $S(\omega)$ is constant for all frequencies (Haykin, 1996):

$$S(\omega) = \sigma^2 \quad \text{for} - \pi < \omega \leq \pi \tag{6.22}$$

6.8.2 Application of WGN to the models

In order to analyse the response of the models to WGN, they have been run, and, in the case of the ANN model, trained and tested, with the sine function due to its representation in the *Frequency Domain.* In Fig. 6.22 the sine function is represented in both the time domain and the frequency domain. The graphs in the frequency domain are plotted on a logarithmic scale on both axis; the frequency is normalised to the range [0, 1].

The chosen period to perform this study covered 2048 data (15-minute time-step) and this number was selected in order to ease the analysis: generating a periodic signal extended over 2^n values, the magnitude of the sine signal, using the discrete fast fourier transform algorithm, can be obtained from Eq. 6.23:

$$|Y| = NA/2 \; = \; (2048)(0.5)/(2) = 512 \tag{6.23}$$

where N is the number of data values, and A the amplitude of the sine wave. The maximum range of frequencies that can be presented in these data series is limited by its discrete condition and the Nyquist Theorem.

The Nyquist Theorem states that the highest frequency which can be accurately represented is less than one-half of the sampling rate. The sampling rate was 1 sample/15 min, hence, the maximum frequency was 1 cycle/30 min, 2 cycle/hr or $5.555x10^{-4}$ Hz. Due to the low values in Herts the frequency was expressed in cycles per hour (cycles/hr). The sine signal generated for this study had 5 cycles on the whole series, that is 5 cycles/512 hr: $9.7656x10^{-}3$ cycles/hr. In normalised frequency, as the maximum frequency corresponds to 1, the sine frequency was located at:

$$\text{Norm Freq}_{sine} = \frac{SineFreq}{MaxFreq} = \frac{9.7656x10^{-3}cycles/hr}{2cycles/hr} \sim 0.005 \tag{6.24}$$

hence, the sine component was at 0.005 as shown in Fig. 6.22, (b), (d), (f) and (h). This signal, because its absence of other components, clearly permitted the observation of the response to WGN by the three models.

WGN was applied to the sine signal at three different levels. In order to measure the relationship between signal and noise the Signal to Noise Ratio (or SNR) in dB has been used according to the following equation:

$$SNR = 10 * \log\left(\frac{\text{Power of sine signal}}{\text{Power of noise}}\right) \tag{6.25}$$

The input signals applied to the conceptual models are shown in Fig. 6.22: the right column is its representation in time domain and the left column is in the frequency domain. (a) and (b) display the pure sine signal; the rest of the plots depict the sine function with a SNR of 30dB in (c) and (d), 20dB in (e) and (f), and 10dB in (g) and (h).

The ANN model has been trained using the flow data from MIKE 11. The input and output data were the results, using the clean sine signal, at chainages 152 and 7929 respectively. Once the model has been trained, it has been tested using, as input data, the results from MIKE 11 at chainage 152 for the three different noise levels. The output generated by the model has been compared to the result obtained by MIKE 11 at chainage 7929.

6.8.3 Results

The input signals shown if Fig. 6.22 have been added to the Glen model. To reduce the influence of the flow on this signal, the chosen period was from 00:15:00 01/FEB/1998 to 08:00:00 22/FEB/1998. This period has a constant and low flow

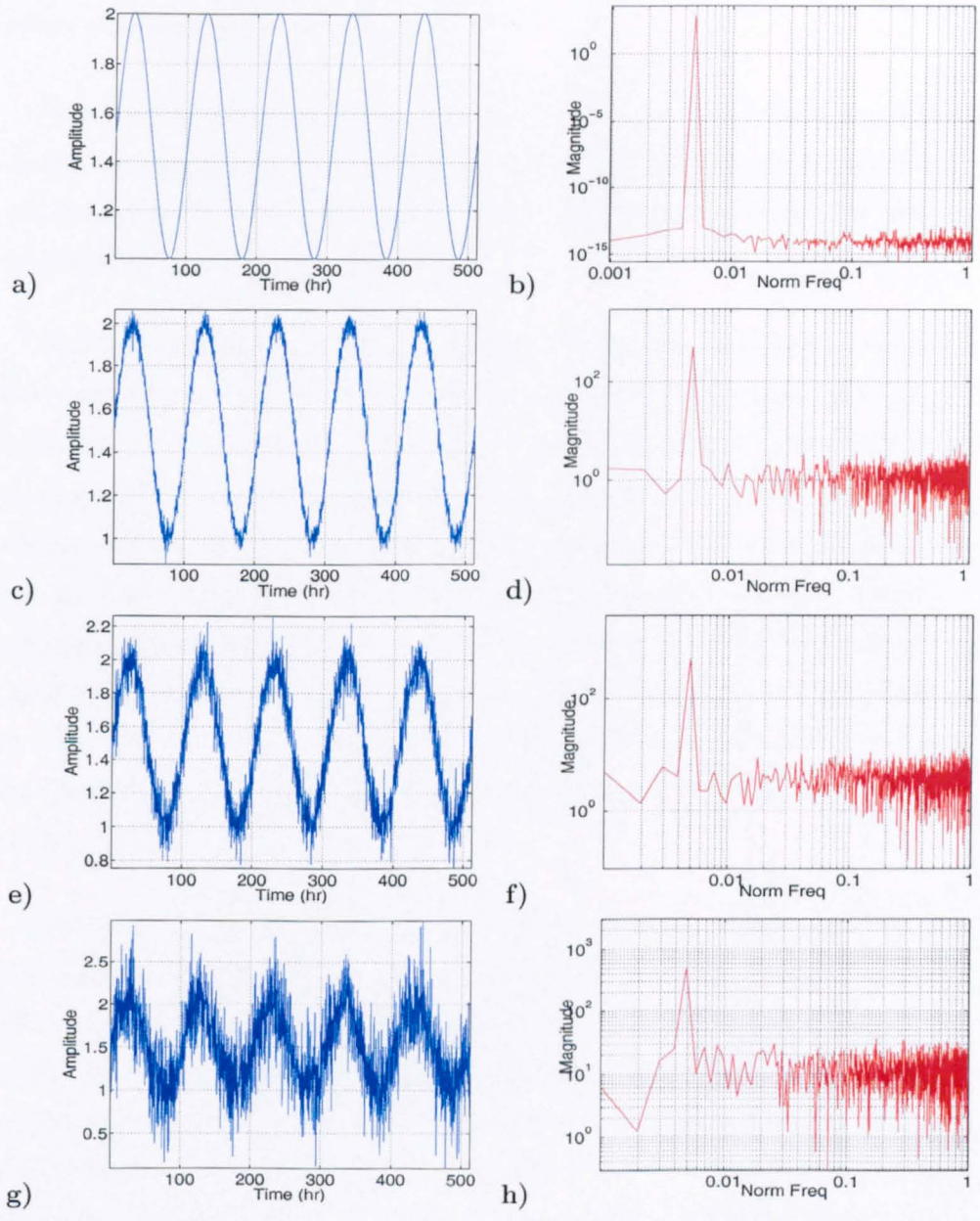


Figure 6.22: Sine signal with different SNR levels: SNR (a) and (b) clean, (c) and (d) 30dB, (e) and (f) 20dB, and (g) and h) 10dB. Left column shows its representation in time domain and right column in frequency domain.

which generated minimum influence on the sine signal. It covers the 2048 span of values explained in the previous section.

The inserted sine signal was added to MIKE 11 and HEC-RAS on the river station (or chainage) 0. The analysed results have been taken from river stations (or chainages) 152 and 7929 which, as indicated before, represented the distance downstream, in meters, from station 0.

The first results presented are from MIKE 11. In order to ease the analysis of the model's behaviour under noise, they are displayed in the frequency domain. Fig. 6.23 plots 8 different graphs showing the outputs of station 152 on the left column and station 7929 on the right column. The plots (a) and (b) represent the sine signal without noise; there is no significant difference between the two signals. In (c) and (d) the noise level is 30dB and the reduction, for a magnitude below 1 came from 0.6 in the normalised frequency in (c), to around 0.2 in (d). The third row had an input SNR of 20dB and, the reduction, for a magnitude below 1 too, came from 0.7 in (e) to around 0.3 in (f). The last row input was with a SNR of 10dB and here, for a magnitude below 1, the reduction came from around 0.8 in (g) to less than 0.4 in (h).

The results from HEC-RAS are displayed in Fig. 6.24. These graphs are presented likewise in the frequency domain. The outputs correspond to station 152 for left column and to station 7929 for right column. The sine component appears, like in the plots for MIKE 11, at 0.005 of the normalised frequency, with a magnitude of 512. The plots (a) and (b) are almost identical, though in plot (b), there is a slightly higher magnitude for the elements after 0.05 of the normalised frequency. In (c) and (d) the noise level is 30dB and the reduction, for a magnitude below 1 came from 0.6 in the normalised frequency in (c), to around 0.3 in (d). The third row had an input SNR of 20dB and, the reduction, for a magnitude below 1 too, came from 0.7 in (e) to around 0.3 in (f). For the last row the input signal had a SNR of 10dB and here, for a magnitude below 1, the reduction came from around

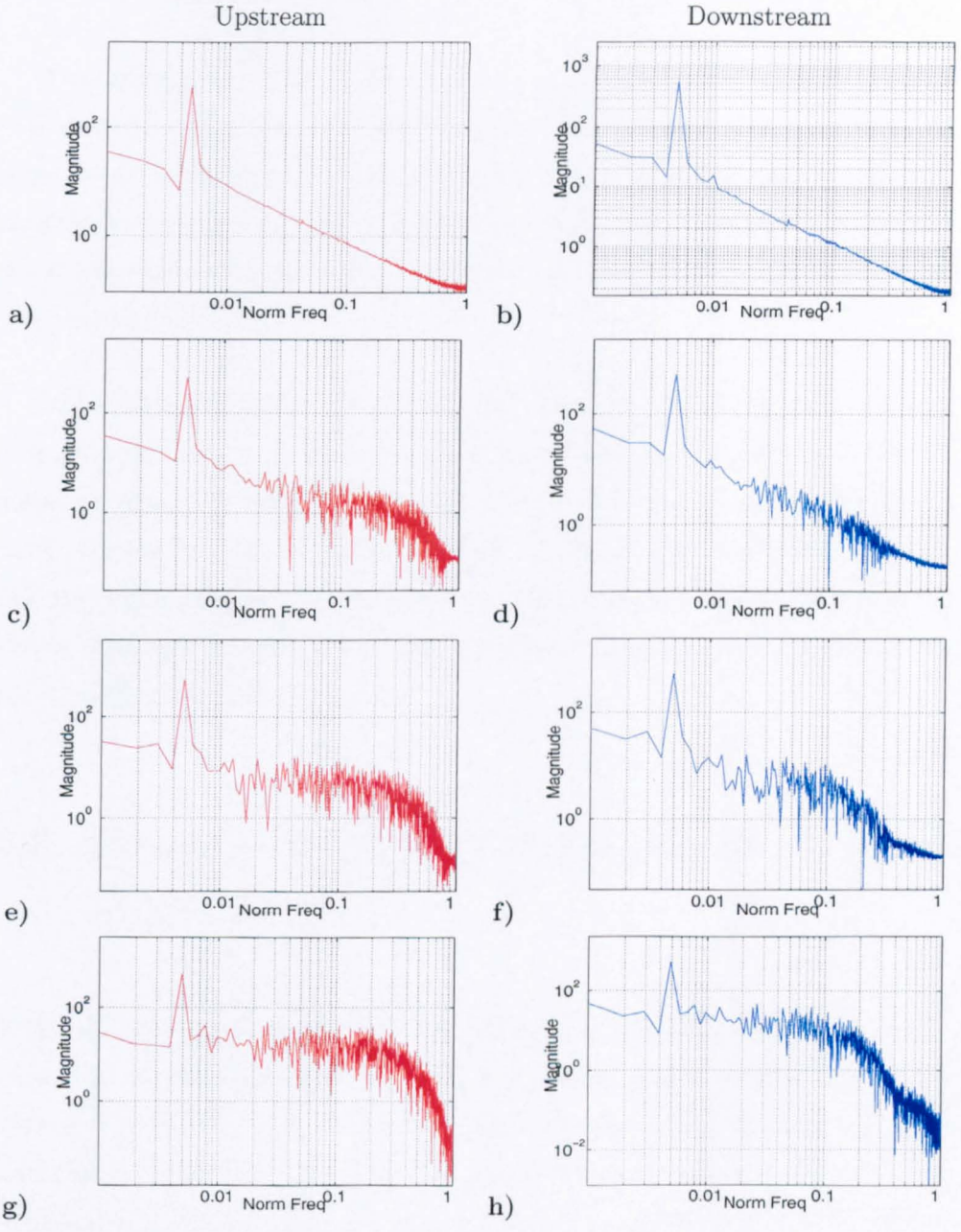


Figure 6.23: Frequency spectrum of MIKE 11 outputs for different SNR levels (a) and (b) clean, (c) and (d) 30dB, (e) and (f) 20dB, and (g) and (h) 10dB. Left column shows river station 152.5 and right column river station 7929.

0.9 in (g) to less than 0.5 in (h).

The simulations from the ANN model are shown in Fig. 6.25. From these results it is clear the model was not as efficient as the conceptual models in reducing the noise level. Comparing this model with MIKE 11, it reduced slightly the noise magnitude in the section [0.04 - 0.3] in the normalised frequency, but increased it in the subsequent section [0.3 to 1]. In all the noise levels the magnitude of the noise was above 1 (10^0.)

Other factor included in the results is efficiency. The efficiency factor has been obtained using the results from the three noise levels against the result from the clean sine signal. The efficiency is plotted in Fig. 6.26. In (a) it is the efficiency for both conceptual models at station 152 when the noise level was still high and hence, the efficiency went down, notoriously at the SNR of 10dB. In (b) the results are at station 7929 and the efficiency values were higher, though the ANN model was the lowest of the three noise levels.

6.9 Response to out of range inputs by the ANN model

Artificial Neural Networks confront a mathematical limitation on simulating processes containing bigger values than those employed during training. There are different options proposed to overcome this limitation: using different activation functions in the last layer, reducing the range to which data series are scaled down, transforming the data series to logarithmic scale, etc., (Imrie et al., 2000; Campolo et al., 1999; Maier and Dandy, 1998; Fine, 1999; Haykin, 1999). In this section the ANN model is presented where the response to out of range data using two different activation functions is considered.

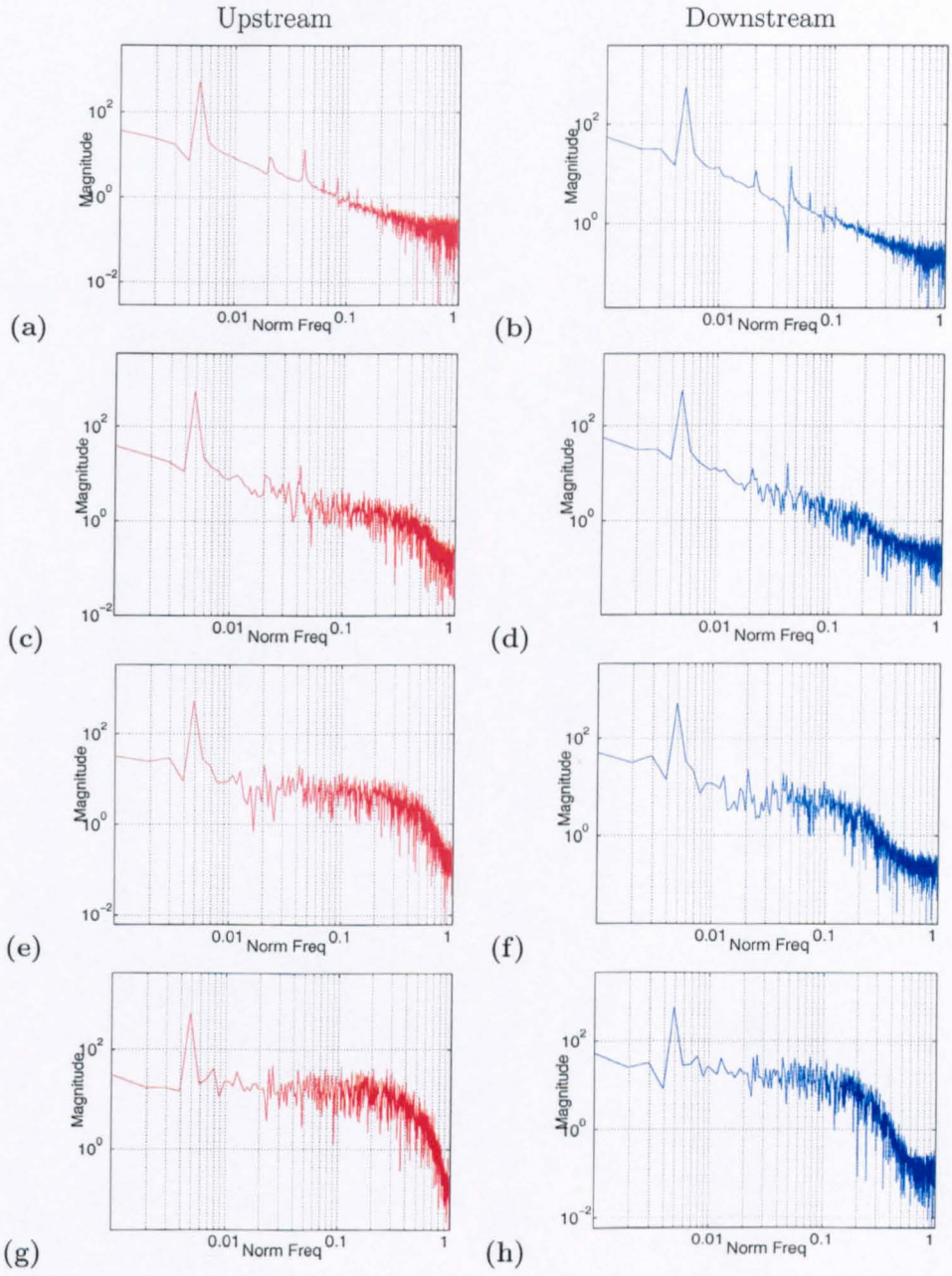


Figure 6.24: Frequency spectrum of HEC-RAS outputs for different SNR levels (a) and (b) clean, (c) and (d) 30dB, (e) and (f) 20dB, and (g) and (h) 10dB.

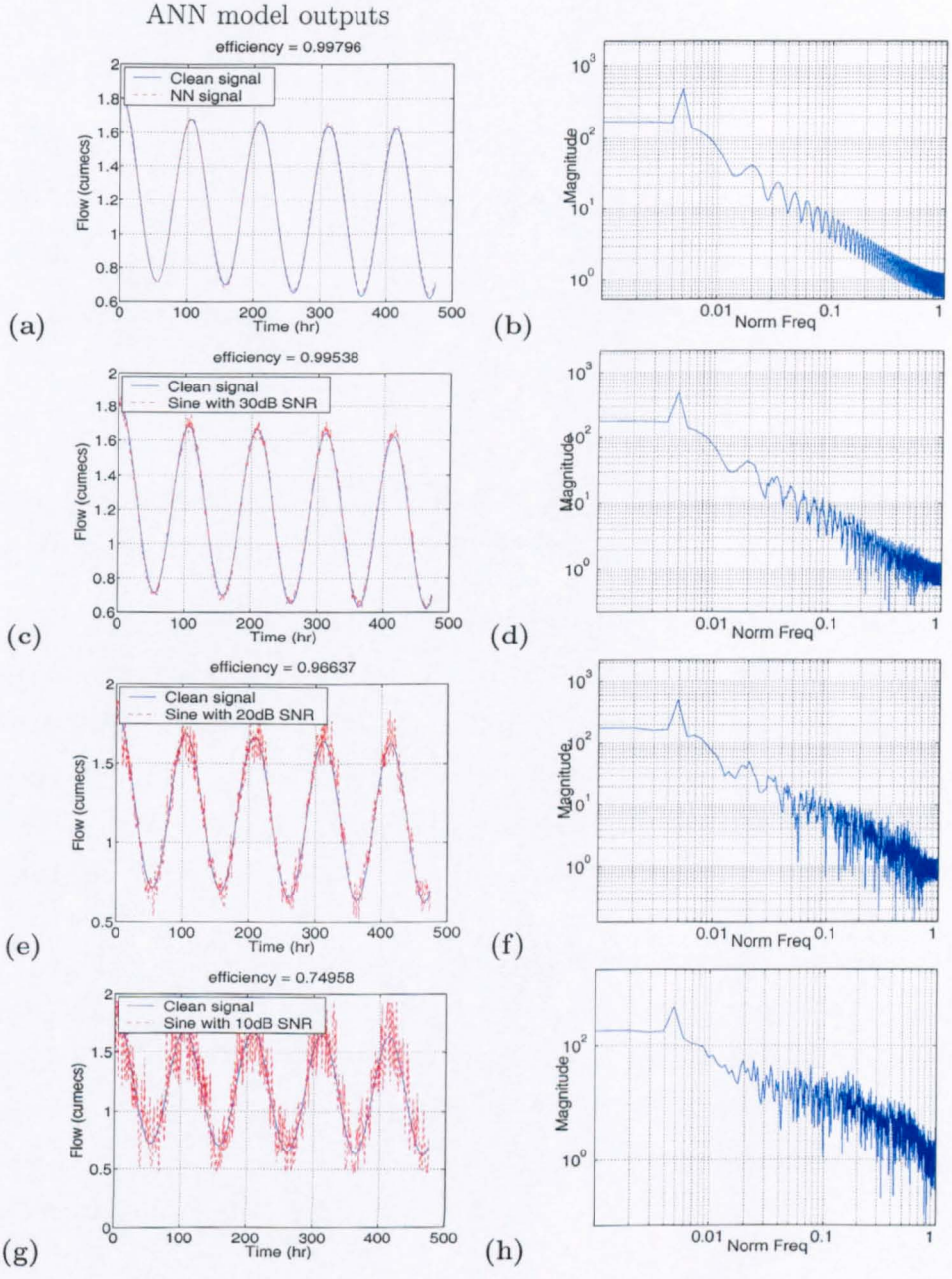


Figure 6.25: ANN model outputs corresponding to the downstream section (7929m). The left column corresponds to the time domain and the right to the frequency domain; each row represents different SNR levels (a) and (b) clean, (c) and (d) 30dB, (e) and (f) 20dB, and (g) and (h) 10dB.

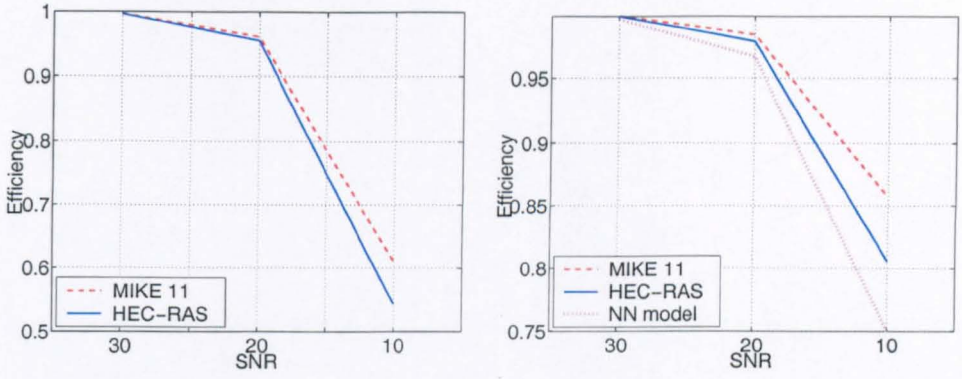


Figure 6.26: (a) Efficiency comparison between MIKE 11 and HEC-RAS for station 152. (b) Efficiency comparison between the three models.

The ANN model was tested with several levels of out of range inputs. The activation functions were changed only in the output layer; that is, in order to maintain the non-linear ability, the ANN model had to keep non-linear functions in the rest of the layers, i.e. the hyperbolic tangent function. The two activation functions under test were the hyperbolic tangent function which has been used throughout all this work, and the linear function (see Fig. 3.5).

The ANN model was the same as that employed to simulate the Upper East Glen river. It has been trained using a sine wave oscillating between the range [2, 3]m^3/s. The data series have been obtained using the MIKE 11 model. The tests have been carried out using the sine wave amplified by 20, 40, 80 and 150 percent of the sine wave that the ANN model was trained with. The results for the hyperbolic tangent function are presented in Fig. 6.27

The hyperbolic tangent function has its outputs limited to the range [-1, 1]. The training data series have been scaled down to the range [-0.8, 0.8], that is, 20% less the maximum output for the function and, nonetheless, the ANN model was not able to simulate even the 20% larger amplified flow signal. As the sine wave was

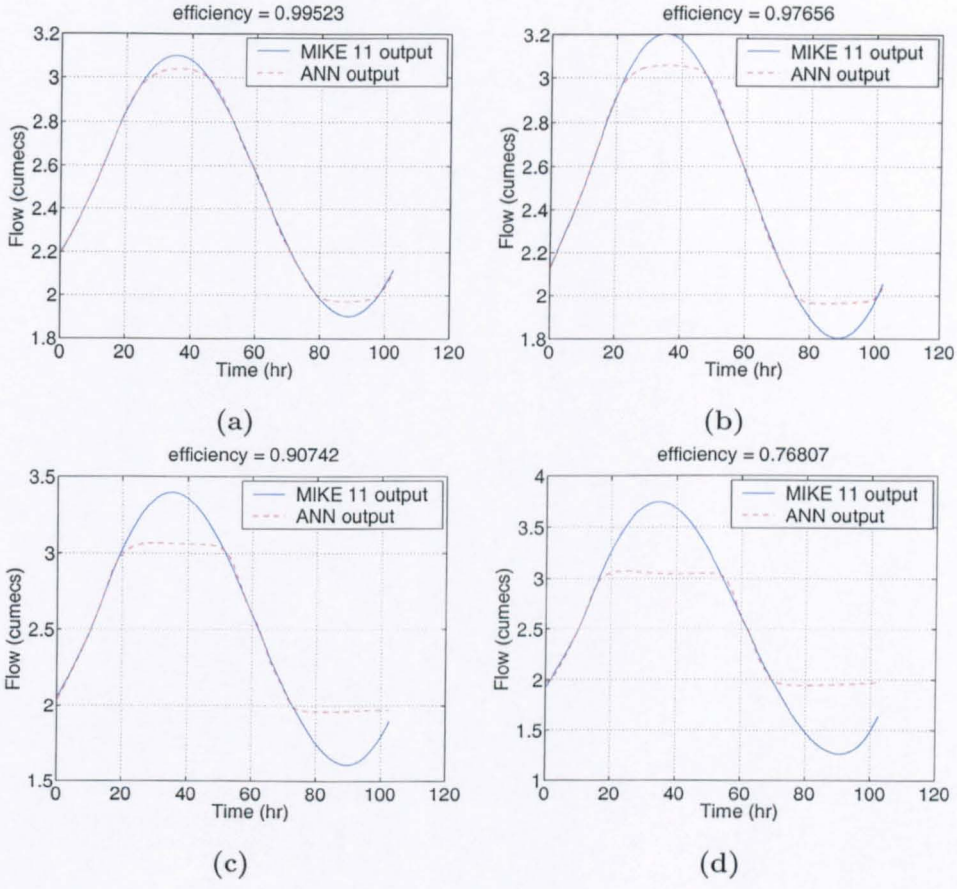


Figure 6.27: ANN model outputs, using the hyperbolic tangent function, to out of range data series: (a) 20%, (b) 40%, (c) 80%, and (d) 150%.

increased, the hyperbolic tangent function bounds became more defined.

The linear function does not have output limits but, as implied by its name, it lacks the non-linear ability. The response to the amplified sine waves is displayed in Fig. 6.28.

With the linear function the ANN model improved its ability to simulate out of range flow and even underestimated low flow. The trade-off of using a linear function is its lower performance in simulating within-range data. To demonstrate this last point, the ANN model was trained and tested, using the linear function in

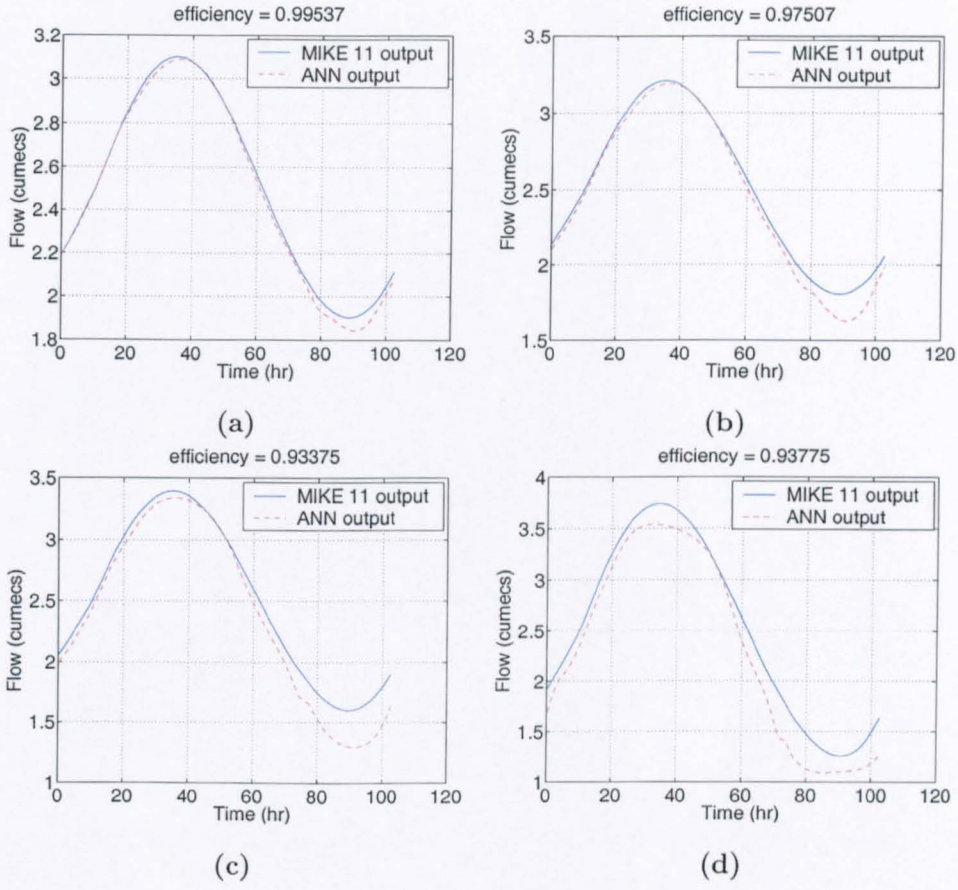


Figure 6.28: ANN model outputs, using the linear function, to out of range data series: (a) 20%, (b) 40%, (c) 80%, and (d) 150%.

its last layer, with the same conditions employed in section 6.6. The result from this test is presented in Fig. 6.29

With the linear function, the ANN model efficiency dropped from 0.9936 with the hyperbolic tangent function to 0.92747 using the linear function.

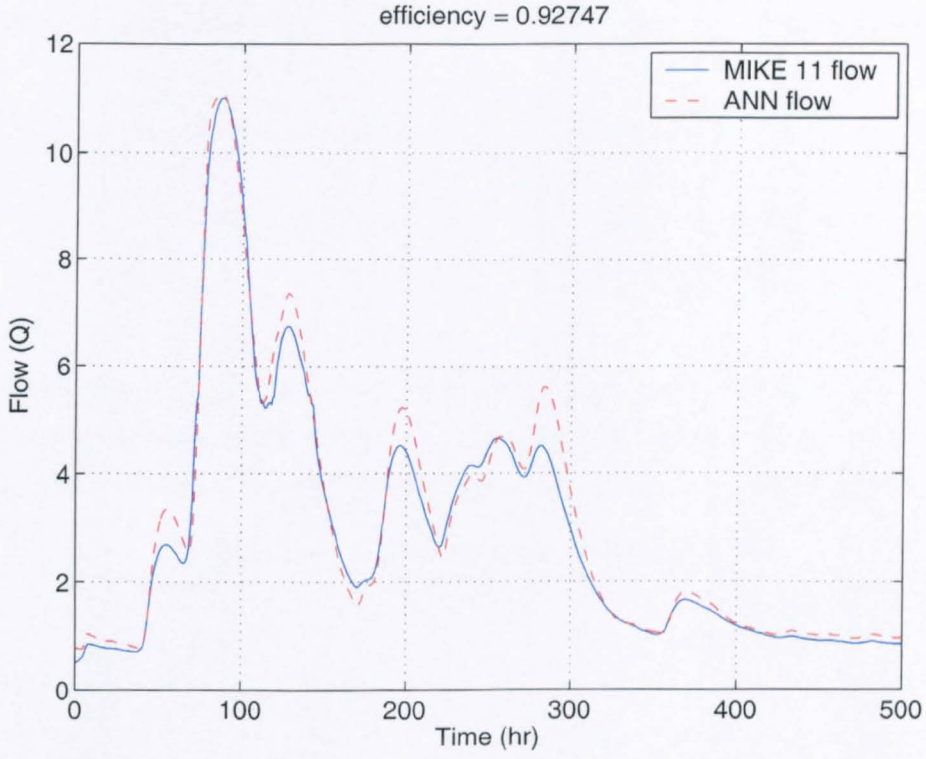


Figure 6.29: ANN model simulation using the linear function in last layer.

6.10 Response to pump-extraction flow by the ANN model

The last analysis performed on the ANN model was simulating a river with its flow affected by pump extraction.

The river Glen was employed to simulate pump extraction by applying a varying level of outflow (with $0.2m^3/s$ minimum flow increments or decrements) at chainage 6377. The rest of the model's conditions remained the same.

To test the model the selected period covered 22 days of January 1998. The specific dates and total hours for every set are described in Table 6.8.

Table 6.8: Training, testing and validation periods for the simulation of the pump extraction

Set (year 1998)	Time (hr)	Start time	End Time
Training	275	01-Jan 00:00	12-Jan 10:45
Validation	118.75	12-Jan 11:00	17-Jan 09:30
Test	118.75	17-Jan 09:45	22-Jan 08:15

Before the flow extraction was applied to the system, the ANN model was trained and tested with the unaffected flow. The resulting hydrograph was plotted in Fig. 6.30. The ANN model matched the river flow with an efficiency $R^2 = 0.99756$.

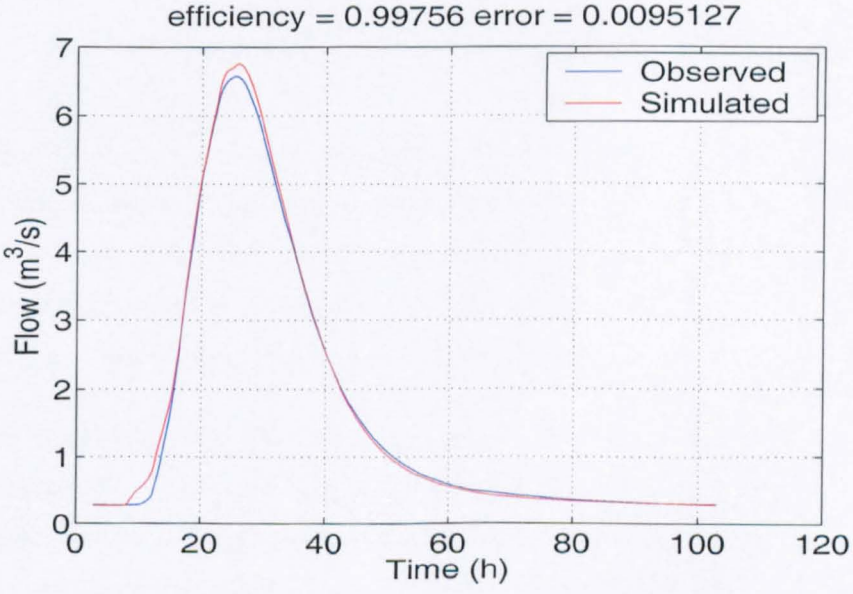


Figure 6.30: ANN model simulation of unaffected flow in the river Glen.

For simulating the pump extraction conditions, the model has been trained using two different approaches: a) adding the pump extraction data like one of the model's inputs and, b) subtracting the pumped flow to the input flow before it is fed to the ANN model. The pump extraction rate is plotted in Fig. 6.31. The highest value

of extraction rate applied to the test period was 34% the flow peak.

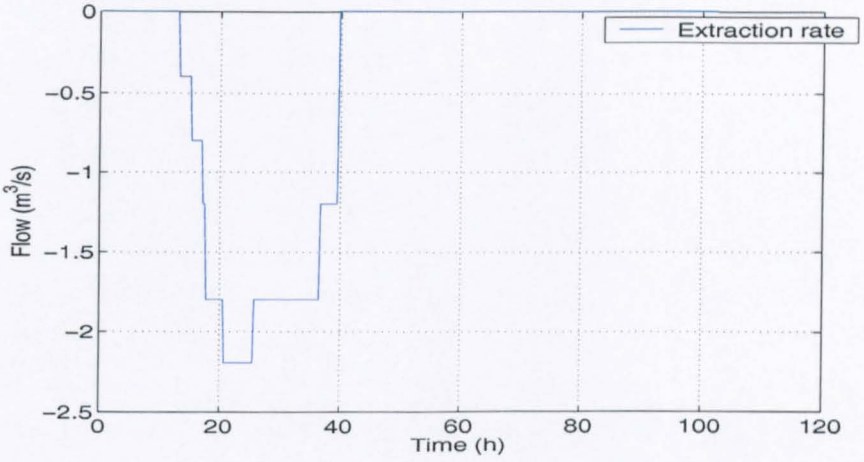


Figure 6.31: Pump extraction rate added to the flow in the river Glen.

Fig. 6.32 shows three plots representing flow at chainage 7929; a) is the ANN model output when no information about the pump extraction is presented to the model; b) presents the ANN model output when the extraction data was fed to the model through a separate input; c) is the ANN model output when the extraction data was subtracted from the input flow at chainage 152.5.

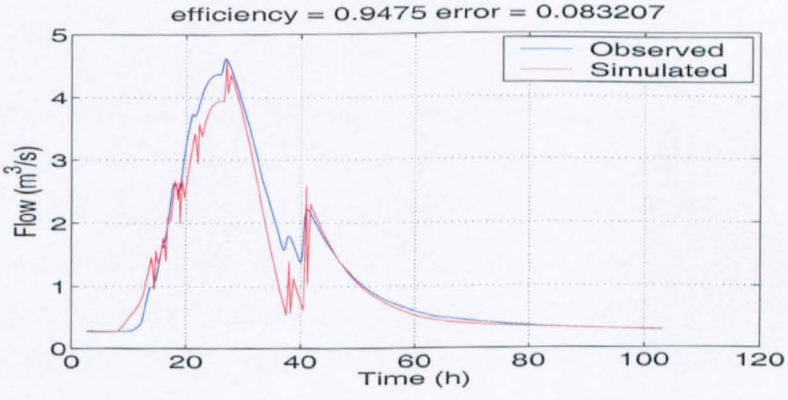
The model simulated the flow at chainage 7929 understandably better, when pump extraction data has been given to the model, rather than when no information has been presented. Nevertheless, the results were lower than, for example, those obtained in simulating a river junction (section 6.7) in which case there were also two independent flow inputs.

6.11 Discussion

HEC-RAS and MIKE 11 are based on the Saint-Venant equations. Both programs have intuitive and conveniently designed graphic user interfaces and ease the models

(a)

(b)



(c)

Figure 6.32: ANN model outputs for period described in Table 5.4. (a) without using pump data, (b) independent input for pump data, (c) adding pump data to input flow.

set-up through several useful features. They can generate accurate estimates of river and channel flows; notwithstanding that the estimation of critical and supercritical flow were poorly performed by HEC-RAS but, as mainly subcritical conditions are found in real situations, this factor can often be ignored.

Modern technology facilitates geographical and hydrological data surveying but a huge amount of effort and expertise is still required to build up a conceptual model and to keep it running and updated. The Welland and Glen model, for example, involved 5 companies and 2 years of work. Therefore, the need for models that facilitate the task of generating reliable flow estimations is still a present quest. That was the reason for presenting and analysing the capabilities and limitations of the ANN model: like an option or complement to conceptual models.

The ANN model performance of the river Glen was highly acceptable with efficiency values R^2 above 0.99 for simulating the MIKE 11 results. Though the modelling was only for a 2-hour span between the two measurement points, it is important to note the fact that input flow from overland runoff was added along the river extension. This lateral inflow increased the difficulty in modelling the system. There were also five structures along this branch which introduced further complications.

The ANN model was capable of simulating, likewise, the downstream flow coming from a junction of a channel and a river. The efficiency R^2 for the period tested was, equally, above 0.99. These two processes would allow the building of a river system network based on the ANN model. If the system involves artificial influences like the pump extraction described in this chapter, the ANN model performance showed a reduction of the efficiency to a value $R^2 = 0.9475$. This decrease in the model's accuracy must be attributed to the model's difficulty to cope with abrupt flow changes as can be seen in Fig. 6.32 a) and b). Nevertheless, the RMSE is, in the case represented by b) in such figure, just 0.083, which, considering that the extraction rate was as high as 34% of the peak flow, does not deviate significantly

from the observed flow.

Sections 6.8 and 6.9 analysed some of the capabilities and limitations experienced by artificial neural networks. Though the ANN model was able to cope with a certain amount of noise, reducing its magnitude, it was not as efficient as the conceptual models. In the case of out of range inputs, in using a specific activation function, it looks like a trade off between accuracy in simulating within the training range and the ability to simulate beyond that training range. To surmount this apparent dichotomy, there are several proposals in the literature (see, for example, Imrie et al. (2000)), and other possibility could be the implementation of a double output ANN model, one for accurate within-range outputs and another one, less precise for out-of-range outputs.

Chapter 7

Conclusions and further work

Nature does not reveal her mysteries once and for all.

Seneca, *Natural Questions*, Book 7

7.1 Conclusions

The need of accurate and reliable flood forecasts forces governmental agencies to invest considerable resources in implementing the required systems. For example, the Environment Agency invested £1.33 million in a contract to develop an integrated flow forecast modelling system (AFFMS) by a consortium of WS Atkins Consultants Ltd, The Danish Hydraulic Institute and Litton Weather Services International. This system is being proved in two catchments in the Anglian Region: the Welland-Glen, and the Witham.

Although these complex deterministic conceptual models for flood forecasting cover highly critical and populated areas, there are rural and small catchments which can be covered by less sophisticated systems giving warnings on extreme events.

The model developed in this thesis has shown the possibilities of a so called "black-box" model based on artificial neural networks and genetic algorithms.

Artificial Neural Networks are an appealing tool because of their ability to learn and adapt to change. They are still evolving and undoubtedly will become a robust and reliable tool in many more fields than those in which they are already successfully applied.

The analyses performed in this thesis shed light on the strengths and weaknesses of this tool in hydrological applications. The use of a filter for rainfall data to remove high frequency elements has been the most important contribution to improve the model's performance. The selection of IIR filters over FIR filters was based on the number of the coefficients needed. Since the number of coefficients is directly related to the phase delay generated by a filter on the signal, the fewer coefficients required by an IIR filter was the decisive factor. The number of coefficients had to be adapted according to the conditions where it was employed. The selected IIR filter had 18 coefficients when applied to the Brue catchment data and improved the ANN model's efficiency (R^2) from just around 0 to above 0.8. For its application to the Bird Creek catchment, the coefficients were only 4.

The other preprocessing options, although not always addressed in the literature, have been also fundamental to the model's proper operation. The random weight initialisation implies the need to repeat the process to find a suitable set and hence, the statistical analysis presented by Iyer and Rhinehart (1999) on the required number of training repetitions proved to be in practice very useful; according to this criterion the number of training repetitions was set to 22.

Genetic algorithms and derived tools have been widely applied to optimisation problems. Implementing a genetic algorithm to generate the optimal parameters of the model was a decisive choice due to the multiple-minima error surface presented by the ANN model (see Fig. 4.10). Using a GA also to define the moving average

rainfall was a determinant factor which allowed the model to respond adequately to events with the same rainfall amount. For comparison, the TF model which has a fixed response to the same amount of rainfall, generates hydrographs which sometimes differ significantly from the observed flow. The response from the two models can be seen in Figs. 3.24, 3.25, 3.26, and 3.27.

The utilization of weather radar rainfall generated less accurate estimations than the raingauge estimations. Three different implementations of the radar data, named *average*, *matrix and average-values*, and *matrix*, were tested. None of the three implementations produced a dominant performance, interchanging the highest performance according to the different periods tested; however all of them fell below the efficiency obtained using raingauge estimations. For example, in the first period tested the highest efficiency obtained with radar data was 0.827, using the *average* implementation. On the other hand, the efficiency using raingauge data was 0.936. Nevertheless weather radars are still improving and will become a reliable source of rainfall estimation. There are several research topics intended to overcome their actual limitations: bright band removal, surmounting attenuation, using double polarisation, etc., but in the meantime, the use of raw polar data instead of processed cartesian data like the rainfall estimations used in this thesis, is an interesting topic to explore and one of the intended further works.

The exploration of simulating a river network by the ANN model, based in comparing the model with two conceptual models generated promising results. The ANN model was not as efficient as the conceptual models in removing noise from the input data but in simulating downstream flow feeding the model with upstream flow produced efficiency values above 0.99. The simulation of a river junction also exhibited high efficiency values. The results for pump extraction were less promising, with an efficiency of 0.94 for the best method, due to the abrupt change in flow extraction. The analysis of out of range inputs established that activation functions limiting the output range restrain the prediction of events outside the calibration range and a linear activation function, which does not limit the output range, lacks

the accuracy in events inside the calibration range.

The two conceptual models, although based on the Saint Venant equations, produced variations in the results for the same river and flow data. The most obvious difference was in simulating flow under critical and supercritical conditions in which cases the HEC-RAS model generated the most discordant results with the calculations made using hydraulic equations.

7.2 Further work

The objective of this thesis was to explore the development of a "flood forecasting model in real-time which utilises ANNs". The core of the model was properly developed but two points need to be taken into account:

Efficiency levels The rainfall-runoff relationship simulated by the ANN model generated efficiency values or error estimations that need to be assessed for forecast applications.

As indicated in this thesis, a rainfall-runoff relationship is affected by several factors whose real influence cannot be extracted, at least completely, from rainfall and runoff data only. Hence, the efficiency level is a factor which would determine the possible application extent of the model.

Selection of training periods The selection of the best suited training periods has been performed by visually choosing those periods covering the highest and lowest available flows according to the rainfall rate of the season involved.

In the UK raw polar data (data collected directly from the radars) from the weather radars are stored only at the radars' location. Radar data sent to the central office have been processed already. Although it is difficult to get access to

these raw polar data, a primary objective is to test the model with such data which will not have undergone the conversion from polar to cartesian coordinate systems. This conversion is in effect a low pass filter operation and the information content of the basic data will therefore be modified.

When consulting forecasters and hydrologists consider artificial intelligence techniques as forecasting tools they are generally reluctant to adopt them, raising several objections but citing mainly -how to trust a "black-box" system when one doesn't know what's inside.

This understandable resistance and the study and analysis of models based on artificial intelligence techniques and conceptual deterministic processes, will provide the next area of research which will be centered on hybrid systems. These hybrid systems may have the advantage to be physically interpreted and to use the ability of artificial neural networks to extract the nonlinear relationships inherent in hydrological systems. The guiding principles will be

- The principle of parsimony. It does not exclude the analysis and possible integration of variables which could have a significant relationship to other variables into the models. Some of these variables can be easily collected; for example, to interpret wet and dry periods a possible variable would be the average temperature. For a forecasting purpose the use of previous values could be an acceptable option since, for example, a monthly temperature chart for Central England shows no significant divergence between the 20-year average and 2003 plots, see Fig 7.1.

- Easy implementation. To prevent the user facing endless adjustments to parameters in order to run the model properly. The user does not need to be an expert in artificial intelligence or in hydrology to deal with all the possible variables, nonetheless must be able to access or visualise them.

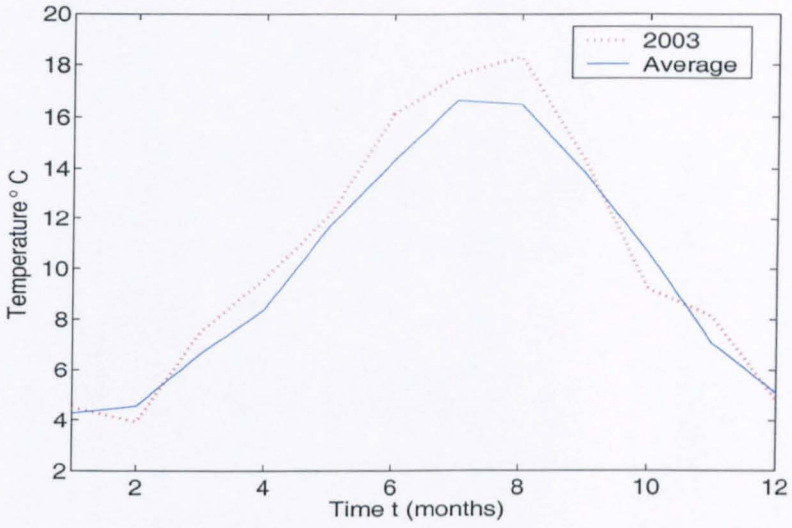


Figure 7.1: Monthly temperatures for Central England: a) 1984-2003 average, b) 2003. Source: The Environment Agency

- Computationally efficient. Although costs of computational power keep cutting down, the program must be able to run several models in parallel or like a background process.
- Forecasts must be accurate for events outside the calibration range in which the model parameters were determined.

The two starting approaches to the hybrid systems will be the tank model for rainfall-runoff relationships, and the Saint-Venant equations for flow-flow relationships.

The tank model will be dynamically adjusted by an artificial neural network. The analysis could be performed on whatever combination produces the best efficiency among the three possible levels: surface storage, lower zone storage, and groundwater storage (DHI, 2000). Fig. 7.2 shows a diagram of the possible structure of the model.

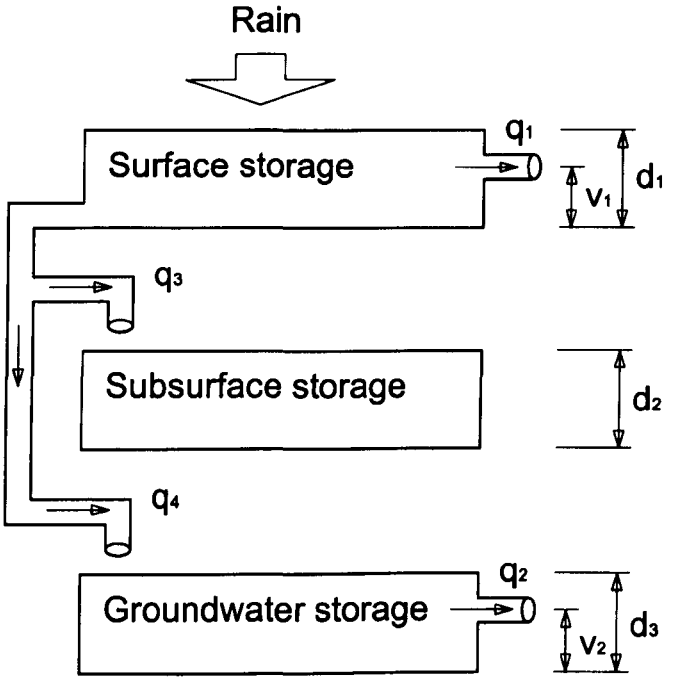


Figure 7.2: Diagram of the hybrid system for rainfall-runoff relationships.

Two possible implementations of the artificial neural network could also be studied: a) adjusting the tanks capacity (d_1, d_2, d_3, v_1, and v_2), the interflow among them (q_1, q_2, q_3 and q_4) or both; b) working with the model's output to minimise the error.

The Saint-Venant equations, as defined by Eqs. 6.1, 6.2 and 6.3 could be solved only for the points concerned, namely upstream flow-downstream flow and the unknown variables A, q_l, S_f, R, and α would be specified dynamically by the artificial neural network.

Finally, the pro's and con's of utilising ANNs having been explored in the context of some of the key hydraulic/hydrological forecasting requirements need to be compared with other sophisticated AI approaches such as support vector machines (SVMs), relevance vector machines (RVMs) and real-time expert systems which

utilise fuzzy modelling concepts.

References

Abrahart, R. J., L. See, and **P. E. Kneale** (1999) Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model. *Journal of Hydroinformatics*, 1(2):103–114.

Ahmad, S. and **S.P. Simonovic** (2001) Developing runoff hydrograph using artificial neural networks. In D. Phelps and G. Sehlke, editors, *Bridging the Gap: Meeting the World's Water and Environmental Resources Challenges. Proceedings of the World Water and Environmental Resources Congress. Orlando, FLA.* Environmental and Water Resources Institute os ASCE.

ASCE (2000a) Artificial neural networks in hydrology. ii: hydrologic applications. *Journal of Hydrologic Engineering*, 5(2):124–137. ASCE task committee on application of artificial neural networks in hydrology (Rao Govindaraju).

ASCE (2000b) Artificial neural networks in hydrology. ii: preliminary concepts. *Journal of Hydrologic Engineering*, 5(2):115–123. ASCE task committee on application of artificial neural networks in hydrology (Rao Govindaraju).

Ashby, W. R. (1960) *Design for a brain: the origin of adaptive behaviour*. Chapman & Hall, London, second edition.

Atlas, D. and **C. W. Ulbrich** (1990) Early foundations of the measurement of rainfall by radar. In D. Atlas, editor, *Radar in meteorology: Battan memorial and 40th Anniversary Radar Meteorology Conference*, pages 86–97. European Commission Publications.

Austin, G. L. (1998) History of radar and radar meteorology. In G L Austin R J Griffith, I D Cluckie and D Han, editors, *Radar Hydrology for Real Time Flood Forecasting*, pages 3–13. European Commission Publications.

BADC (2004) British atmospheric data centre. Http://badc.nerc.ac.uk/data/mst/.

Battan, L.J. (1973) *Radar observation of the atmosphere*. University of Chicago Press, Chicago.

Bernard, W., D.E. Rumelhart, and **M.A. Lehr** (1994) Neural networks: applications in industry, business and science. *Communications of the ACM*, 39:93–105.

Bertsekas, D.P. and **J.N. Tsitsiklis** (1996) *Neuro-dynamic Programming*. Athena Scientific, Belmont, Mass.

Bezdek, J. C. (1993) Fuzzy models - what are they, and why? *IEEE Transactions on Fuzzy Systems*, 1(1):1–6.

Bosch, J. M. and **J. D. Hewlett** (1982) A review of catchment experiments to determine the effect of vegetation changes on water yield and evapotranspiration. *Journal of Hydrology*, 55:3–23.

Campolo, M., P. Andreussi, and **A. Soldati** (1999) River flood forecasting with a neural network model. *Water Resources Research*, 35(4):1191–1197.

Chadwick, A. J. and **J. Morfett** (1993) *Hydraulics in civil and environmental engineering*. Chapman & Hall, London, second edition.

Cheng, B. and **D.M. Titterington** (1994) Neural networks: A review from a statistical perspective. *Statistical Science*, 9(1):2–54.

Chérif, R., J.-L. Robert, and **R. Lagacé** (2002) Genetic algorithm calibration of hydrologic model. In I. D. Cluckie, D. Han, J. P. Davis, and S. Heslop, editors, *Proceedings of the fifth international conference on Hydroinformatics, Cardiff, UK*, pages 1417–1422. IWA Publishing.

Cichocki, A. and **R. Unbehauen** (1993) *Neural Networks for Optimization and Signal Processing*. John Wiley & Sons, Chichester.

Cluckie, I.D., R.J. Griffith, A. Lane, and **K.A. Tilford** (2000) Radar hydrometeorology using a vertically pointing radar. *Hydrology and Earth System Sciences*, 4(4):565–580.

Cole, H.W. (1992) *Understanding radar*. Blackwell Scientific Publications, Oxford, second edition.

Coley, D. A. (1999) *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific Publishing Co. Pte. Ltd, Singapore.

Collier, C. G. (1996) *Applications of weather radar systems: a guide to uses of radar data in meteorology and hydrology*. John Wiley & Sons Ltd, Chichester, second edition.

Collinge, V. K. and **C. Kirby** (1987) The development of weather radar in the united kingdom. In V. K. Collinge and C. Kirby, editors, *Weather radar and flood forecasting*, pages 3–18. Wiley, Chichester.

Cooley, R., B. Mobasher, and **J. Srivastava** (1997) Web mining: Information and pattern discovery on the world wide web. Technical Report TR 97-027, University of Minesota, Dept. of Computer Science, Minneapolis.

Costa, M. H., A. Botta, and **J. A. Cardille** (2003) Effects of large-scale changes in land cover on the discharge of the tocantins river, southeastern amazonia. *Journal of Hydrology*, 283:206–217.

Coulthard, T. J., M. J. Kirkby, and **M. G. Macklin** (1997) Modelling hydraulic, sediment transport and slope processes, at a catchment scale, using a cellular automaton approach. In R. T. Pascoe, editor, *Proceedings of the second annual conference: GeoComputation 97*, pages 309–318. University of Otago, Dunedin, New Zealand.

Dawson, C. W. and **R. Wilby** (1998) An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*, 43(1):47–66.

Deco, G. and **D. Obradovic** (1996) *An Information-Theoretic Approach to Neural Computing*. Springer-Verlag New York Inc., New York, first edition.

Deco, G. and **B. Schürmann** (2001) *Information dynamics: foundations and applications*. Springer-Verlag New York Inc., New York, first edition.

DHI (2000) *MIKE 11, A Modelling System for Rivers and Channels: Reference Manual*. DHI Water & Environment.

Dingman, S. L. (1994) *Physical Hydrology*. Prentice Hall, Upper Saddle River, N.J.

Douglas, R. H. (1990) The stormy weather group (canada). In D. Atlas, editor, *Radar in meteorology: Battan memorial and 40th Anniversary Radar Meteorology Conference*, pages 61–68. European Commission Publications.

Dreyfus, H. L. (1992) *What computers still can't do: a critique of artificial reason*. The MIT Press, Cambridge, Mass.

Evgeniou, T., **M. Pontil**, and **T. Poggio** (2000) Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50.

Federico, G. (2000) How did they feed us? the growth of world agricultural output, 1800-1938. Agricultural History Centre Working Paper # 0203, University of California at Davis.

Fernando, D.A.K. and **A.W. Jayawardena** (1998) Runoff forecasting using rbf networks with ols algorithm. *Journal of Hydrologic Engineering*, 3(3):203–209.

Fine, T. L. (1999) *Feedforward Neural Network Methodology (Statistics for Engineering and Information Science)*. Springer-Verlag New York Inc., New York.

Fletcher, J. O. (1990) Early developments of weather radar during world war ii. In D. Atlas, editor, *Radar in meteorology: Battan memorial and 40th Anniversary Radar Meteorology Conference*, pages 3–6. European Commission Publications.

Gautam, D.K. and **K.P. Holz** (2001) Rainfall-runoff modelling using adaptive neuro-fuzzy systems. *Journal of Hydroinformatics*, 03(01):3–10.

Ginsberg, M. L. (1993) *Essentials of Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., San Francisco, first edition.

Goldberg, D. E. (1989) *Genetic Algorithms in search, optimization, and machine learning*. Addison-Wesley.

Golob, R., **T. Štokelj**, and **D. Grgič** (1998) Neural-network-based water inflow forecasting. *Control Engineering Practice*, 6:593–600.

Gwangseob, K. and **A. P. Barros** (2001) Quantitative flood forecasting using multisensor data and neural networks. *Journal of Hydrology*, 246:45–62.

Hajjam, S. (1997) *Real-time flood forecasting model intercomparison and parameter updating using rain gauge and weather radar data*. Ph.D. thesis, University of Salford, Salford, UK.

Han, D. (1991) *Weather radar information processing and real-time flood forecasting*. Ph.D. thesis, University of Salford, Salford, UK.

Han, D., **I. Cluckie**, and **W. Kang** (2002) Flow modelling using relevance vector machine. In I. D. Cluckie, D. Han, J. P. Davis, and S. Heslop, editors, *Proceedings of the fifth international conference on Hydroinformatics, Cardiff, UK*, pages 1429–1434. IWA Publishing.

Harrison, D. L., **S. J. Driscoll**, and **M. Kitchen** (2000) Improving precipitation estimates from weather radar using quality control and correction techniques. *Meteorological Applications*, 6:135–144.

Haykin, S. (1996) *Adaptive filter theory.* Prentice-Hall, Inc., New Jersey, third edition.

Haykin, S. (1999) *Neural Networks - A Comprehensive Foundation.* Prentice-Hall, Inc, New Jersey.

HECHMS (2000) *Hydrologic Modelling System: Technical Reference Manual.* U.S. Army Corps of Engineers, Hydrologic Engineering Center, Davis, CA.

HECRAS (2002a) *River Analysis System: Hydraulic Reference Manual Version 3.1.* U.S. Army Corps of Engineers, Hydrologic Engineering Center, Davis, CA.

HECRAS (2002b) *River Analysis System: User's Manual Version 3.1.* U.S. Army Corps of Engineers, Hydrologic Engineering Center, Davis, CA.

Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems.* Addison-Wesley, New York.

Hornberger, G. M. (1998) *Elements of physical hydrology.* John Hopkins University Press, Baltimore.

Illingworth, A. J., T. M. Blackman, and **W. F. Goddard** (2000) Improved rainfall estimates in convective storms using polarisation diversity radar. *Hydrology and Earth System Sciences*, 4(4):555–563.

Imrie, C.E., S. Durucan, and **A. Korre** (2000) River flow prediction using artificial neural networks: generalisation beyond the calibration range. *Journal of Hydrology*, 233:138–153.

Iyer, M. S. and **R. R. Rhinehart** (1999) A method to determine the required number of neural-network training repetitions. *IEEE Transactions on Neural Networks*, 10(2):427–432.

Jayawardena, A.W. and **N. Muttil** (2004) Application of data-driven approaches for modelling runoff from a steep-sloped catchment in hong kong. In Phoon

& Babovic Liong, editor, *6th International Conference on Hydroinformatics*. World Scientific Publishing Company.

Jones, J. A. A. (1997) *Global Hydrology: processes, resources and environmental management*. Longman, Harlow.

Joss, J. and **A. Waldvogel** (1990) Precipitation measurement and hydrology. In D. Atlas, editor, *Radar in meteorology: Battan memorial and 40th Anniversary Radar Meteorology Conference*, pages 577–606. European Commission Publications.

Keane, A. J. (1995) Genetic algorithm optimization of multi-peak problems: Studies in convergence and robustness. *Artificial Intelligence in Engineering*, 9:75–83.

Kodaira, N. and **J. Aoyagi** (1990) History of radar meteorology in japan. In D. Atlas, editor, *Radar in meteorology: Battan memorial and 40th Anniversary Radar Meteorology Conference*, pages 69–76. European Commission Publications.

Lakhmi, C. J. and **N. M. Martin**, editors (1999) *Fusion of neural networks, fuzzy sets, and genetic algorithms: industrial applications*. CRC Press, Boca Raton.

Lange, N.T. (1999) New mathematical approaches in hydrological modeling-an application of artificial neural networks. *Physics and Chemistry of the Earth*, 24(1):31–35.

Lekkas, D. F., **C. E. Imrie**, and **M. J. Lees** (2001) Improved non-linear transfer function and neural network methods of flow routing for real-time forecasting. *Journal of Hydroinformatics*, 3(3):32–55.

Liberty, J. and **D. B. Horvath** (2000) *Sams Teach Yourself C++ for Linux in 21 days*. Sams Publishing, Indianapolis, Indiana.

Liong, S.Y., **W.H. Lim**, **T. Kojiri**, and **T. Hori** (2000) Advance flood forecasting for flood stricken bangladesh with a fuzzy reasoning method. *Hydrological Processes*, 14:431–448.

Lorup, J. K., **J. C. Refsgaard**, and **D. Mazvimavi** (1998) Assessing the effect of land use change on catchment runoff by combined use of statistical tests and hydrological modelling: Case studies from zimbabwe. *Journal of Hydrology*, 205:147–163.

Luger, G. F. (1998) *Artificial Intelligence. Structures and Strategies for Complex Problem Solving*. Addison-Wesley, Reading, MA, third edition.

Lumb, A. M., **R. B. McCammon**, and **J. L. Kittle** (1994) *Users manual for an expert system (HSPEXP) for calibration of the Hydrological Simulation Program-Fortran*. U. S. Geological Survey, Water-Resources Investigations Report 94-4168, Restion, Virginia.

Mahabir, C., **F.E. Hicks**, and **A. Robinson** (2003) Application of fuzzy logic to forecast seasonal runoff. *Hydrological Processes*, 17:3749–3762.

Maidment, D. R., editor (1993) *Handbook of Hydrology*. McGraw-Hill, New York; London.

Maier, H. R. and **G. C. Dandy** (1998) The effect of internal parameters and geometry on the performance of back-propagation neural networks: an empirical study. *Environmental Modelling & Software*, 13:193–209.

Maier, H. R. and **G. C. Dandy** (2000) Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. *Environmental Modelling & Software*, 15:101–124.

McCulloch, W.S. and **W. Pitts** (1943) A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

Minsky, M. and **S. Papert** (1988) *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge, Mass, expanded edition.

Mitchell, M. (1996) *An Introduction to Genetic Algorithms*. First MIT Press, Cambridge, Massachusetts.

Mitra, S. K. (2001) *Digital signal processing: a computer-based approach*. McGraw-Hill, New York, second edition.

Moore, R. J., D. A. Jones, D. R. Cox, and **V. S. Isham** (2000) Design of the hyrex raingauge network. *Hydrology and Earth System Sciences*, 4(4):523–530.

Nandalal, K. D. W. (2002) An application of genetic algorithms in optimal reservoir operation. In I. D. Cluckie, D. Han, J. P. Davis, and S. Heslop, editors, *Proceedings of the fifth international conference on Hydroinformatics, Cardiff, UK*, pages 1453–1458. IWA Publishing.

Nash, J. E. and **V. Sutcliffe** (1970) River flow forecasting through conceptual models, i. a discussion of principles. *Journal of Hydrology*, 10(3):282–290.

Nicholls, J. G. (2001) *From neuron to brain*. Sinauer Associates, Sunderland, Mass, fourth edition.

Nilsson, N. J. (1998) *Artificial Intelligence: a new synthesis*. Morgan Kaufmann Publishers, Inc., San Francisco, first edition.

Oppenheim, A.V. and **R.W. Schafer** (1975) *Digital signal processing*. Prentice-Hall, New Jersey, first edition.

Özelkan, E.C. and **L. Duckstein** (2001) Fuzzy conceptual rainfall-runoff models. *Journal of Hydrology*, 253:41–68.

Petridis, V., E. Paterakis, and **A. Kehagias** (1998) A hybrid neural-genetic multimodel parameter estimation algorithm. *IEEE Transactions on Neural Networks*, 9(5):75–83.

Petterssen, S. (1958) *Introduction to meteorology*. McGrau-Hill Book Company, Inc., Tokyo, second edition.

Proakis, J. G. and **D. G Manolakis** (1992) *Digital signal processing: principles, algorithms, and applications*. Macmillan, New York, second edition.

Probert-Jones, J. R. (1962) The radar equation in meteorology. *Quarterly journal of the Royal Meteorological Society*, 88:485–495.

Probert-Jones, J. R. (1990) A history of radar meteorology in the united kingdom. In D. Atlas, editor, *Radar in meteorology: Battan memorial and 40th Anniversary Radar Meteorology Conference*, pages 54–60. European Commission Publications.

Ripley, B.D. (1996) *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge.

Rogers, L.L. and **F.U. Dowla** (1994) Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. *Water Resources Research*, 30(2):457–481.

Sajikumar, N. and **B.S. Thandaveswara** (1999) A non-linear rainfall-runoff model using an artificial neural network. *Journal of Hydrology*, 216:32–55.

Simanovic, S. P. (1990) An expert system for the selection of a suitable method for flow measurement in open channels. *Journal of Hydrology*, 112:237–256.

Sivapgragasam, C., **S.Y. Liong**, and **M.F.K. Pasha** (2001) Rainfall and runoff forecasting with ssa-svm approach. *Journal of Hydroinformatics*, 3:141–152.

Smith, N. (1971) *A history of dams*. P. Davies, London, first edition.

Smith, P. (1990) Precipitation measurement and hydrology: panel report. In D. Atlas, editor, *Radar in meteorology: Battan memorial and 40th Anniversary Radar Meteorology Conference*, pages 607–618. European Commission Publications.

Sutton, O.G. (1961) *Understanding Weather*. Penguin Books Ltd, Harmondsworth, first edition.

Swingle, D. M. (1990) Weather radar in the united states army's forth monmouth. In D. Atlas, editor, *Radar in meteorology: Battan memorial and 40th Anniversary Radar Meteorology Conference*, pages 7–15. European Commission Publications.

Tilford, K. A. (1992) *Weather radar data for operational hydrology*. Ph.D. thesis, Salford University, Salford, UK.

Topping, B.H.V., J. Sziveri, A. Bahreinejad, J.P.B. Leite, and **B. Cheng** (1998) Parallel processing, neural networks and genetic algorithms. *Advances in Engineering Software*, 29(10):763–786.

Van Dijk, A. I. J. M. and **L. A. Bruijnzeel** (2001) Modelling rainfall interception by vegetation of variable density using an adapted analytical model. part 1. model description. *Journal of Hydrology*, 247:230–238.

Varas, E. and **M. von Chrismar** (1995) Expert system for the selection of methods to calculate design flood flows. *Hydrological Sciences Journal*, 40(6):739–750.

Ward, R. C. and **M. Robinson** (1990) *Principles of Hydrology*. McGraw-Hill International (UK) Limited, Berkshire, third edition.

WEMRC (2004) Water and environmental management research centre. Http://www.cen.bris.ac.uk/research/wemrc/welcome.htm.

Williamson, J. (2004) A dynamic interaction between machine learning and the philosophy of science. To appear in Minds and Machines.

Winston, P. H. (1992) *Artificial Intelligence*. Addison-Wesley, third edition.

WMO (1992) Simulated real-time intercomparison of hydrological models. Operational hydrology eport 38, World Meteorological Organization.

Wratt, D. S., R. N. Ridley, M. R. Sinclair, H. Larsen, S. M. Thompson, et al. (1996) The new zealand southern alps experiment. *Bulletin of the American Meteorological Society*, 77(4):683–692.

Wurbs, R. A. (1995) *Water Management Models: A Guide to Software*. Prentice Hall PTR, New Jersey, first edition.

Yang, C.C., C.S. Chen, and **Chang L.C.** (2004) The interval estimation of parameters for neural network on flood forecasting. In Phoon & Babovic Liong, editor, *6th International Conference on Hydroinformatics*. World Scientific Publishing Company.

Yang, Z. (1999) *Application of All-Pole Physically Realisable Transfer Function Models to Flow Simulation and Real-Time Flood Forecasting*. Ph.D. thesis, University of Bristol, Bristol, UK.

Appendix A

Flow Diagram

A flow diagram has been used to describe the algorithms employed in the ANN model. This flow diagram has been split up to four parts to ease looking at it. The variables present in the four diagrams are described in the following list.

- Max Epoch = Maximum number of epochs is set to 500
- r = rainfall series
- f = runoff series
- $\tau 1$ = rainfall values to feed the ANN at ith step
- $\tau 2$ = average rainfall values
- x_i = ith input vector
- o_i = ith output vector
- y_i = ith target (or desired) vector
- N_T = number of iterations for the training period

- N_V = number of iterations for the validation period
- N_{TE} = number of iterations for the testing period
- η = variable learning rate
- W = matrix of weights
- β = vector of biases
- W_{back} = Backup matrix of weights. It saves the best set of weights.
- β_{back} = Backup vector of biases. It saves the best set of biases.
- e_{vp} = variable used to keep the lowest error in the evaluation section.
- e_T = error generated using the test data.

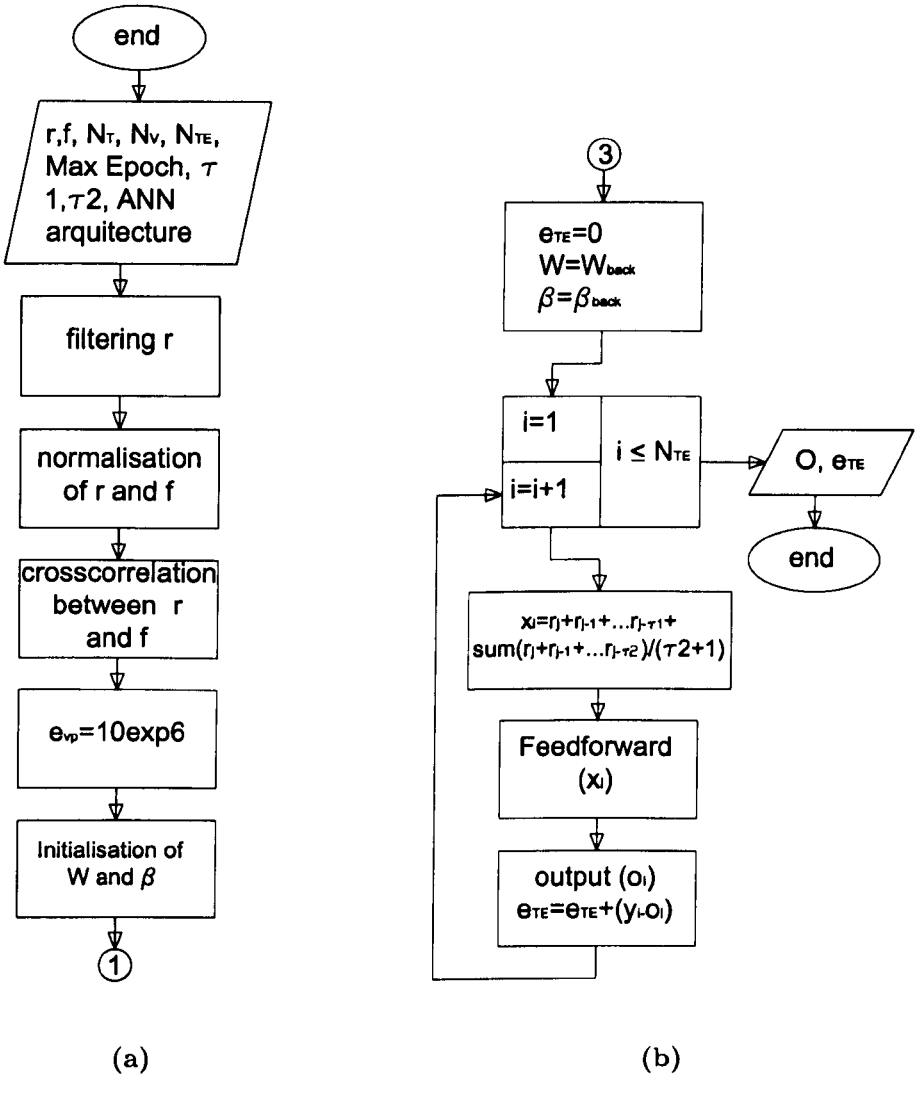


Figure A.1: Diagrams representing the ANN training algorithm: (a) starting section, (b) ending section

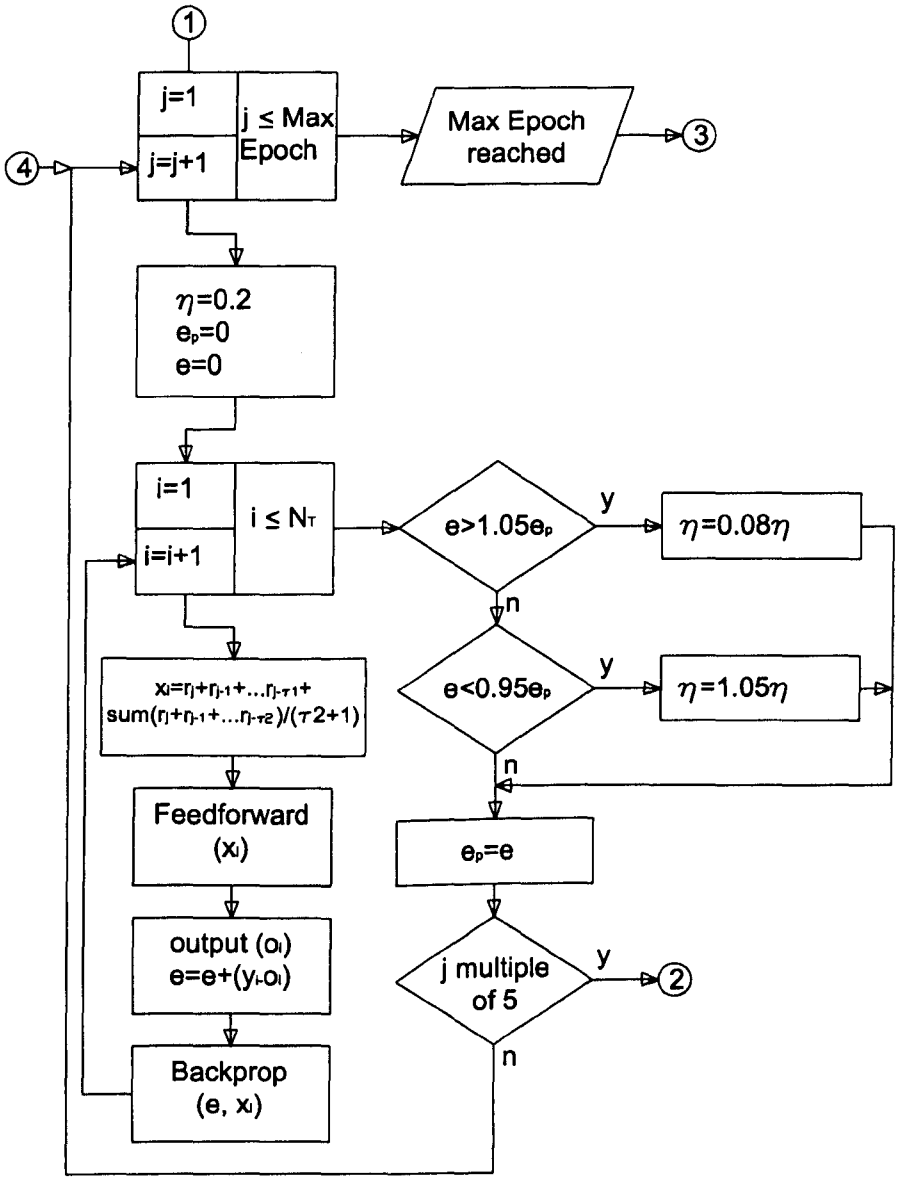


Figure A.2: Diagram representing the ANN training algorithm

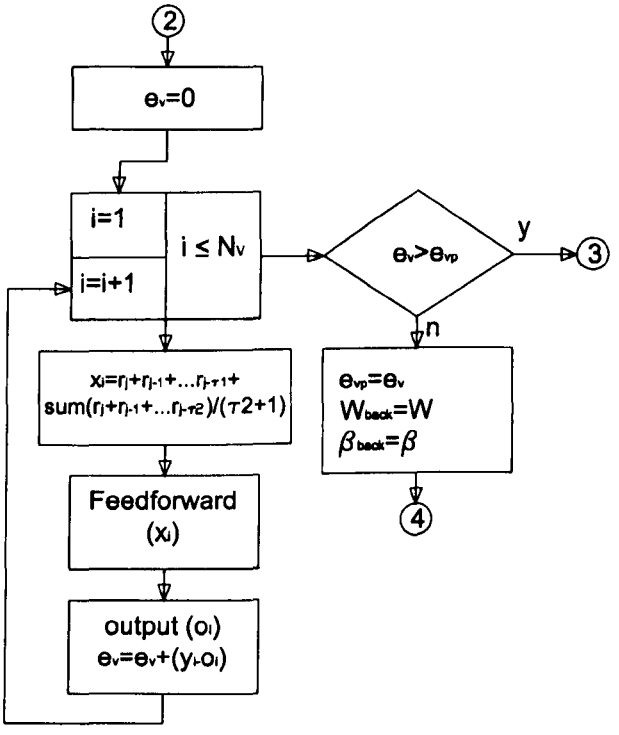


Figure A.3: Diagram representing the stopping criteria of the ANN training algorithm

Appendix B

Transfer Function Model

B.1 Introduction

A transfer function (TF) model is a linear system in the form of difference equations which is characterised by its response to a unit impulse. The analysis, design, and characterisation of a TF model in the time-discrete domain is greatly eased by the z-transform which is defined previous to presenting the TF model.

B.2 The z-transform

The z-transform is a basic tool in the analysis of discrete-time signals and linear time invariant systems. It allows insight into the transient behaviour, the steady state behaviour, and the stability of discrete-time systems. The z-transform of a discrete-time signal $u(t)$ is mathematically defined by

$$U(z) \equiv \sum_{n=-\infty}^{\infty} u(t) z^{-t} \tag{B.1}$$

where z is a complex variable. Below are listed its most important properties. A more complete list of z-transform properties are described in Proakis and Manolakis (1992); Mitra (2001); Oppenheim and Schafer (1975).

- Linearity: for any constants a_1 and a_2

$$a_1 U_1(z) + a_2 U_2(z) \iff a_1 u_1(t) + a_2 u_2(t) \tag{B.2}$$

- Time delay: Multiplication by z^{-k} means a time delay of k time-steps

$$z^{-k} U(z) \iff u(t-k) \tag{B.3}$$

- Time advancing: Multiplication by z^k means time advancing by k time-steps:

$$z^k U(z) \iff u(t+k) \tag{B.4}$$

If u(t) is to have a z-transform, then U(z) must be finite. The set of values of z for which U(z) is finite is known as *region of convergence* (ROC). As there is no unique relationship between a sequence and its z-transform, the z-transform must always be specified with its ROC.

In the case of linear time invariant systems, like the transfer function model concerned with in this thesis, their z-transforms are rational functions of z^{-1},

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \ldots + a_N z^{-N}} \tag{B.5}$$

This equation can be written in factored form as

$$H(z) = z^{N-M} \frac{\prod_{k=1}^{M} (z - z_k)}{\prod_{k=1}^{N} (z - p_k)} \tag{B.6}$$

hence, a z-transform has M finite zeros at $z = z_1, z_2, ..., z_M$ and N finite poles at $z = p_1, p_2, ..., p_N$. It also follows from the above equation that there are additional (N-M) zeros at $z = 0$ if $N > M$ or additional (M-N) poles at $z = 0$ if N¡M. The location of the poles on the ROC determines the stability property of the system. The analysis of the poles and zeros of the transfer function model is presented in the following section.

B.3 The transfer function model

The transfer function model simulates only flow peaks; hence, a separation between flow peaks or quick flow and base or slow flow has been performed on the total flow. The base flow has been added after the simulation to obtain the total flow and compare it to the observed flow. The method employed in the separation between quick and slow flows is presented first as part of the transfer function model.

B.3.1 Base flow separation

The separation of base flow from 'quick flow' has been done using the approach proposed by Hewlett and Hibbert (Ward and Robinson, 1990) in which a line of constant slope, defined by Eq. B.7, is projected from the beginning of the rise limb to the point where it intersects the falling limb of a hydrograph.

$$slope = 0.000546 m^3/s/(km^2 h) \tag{B.7}$$

Applying Eq. B.7 to the Brue catchment with an area of 135 km^2 and a 15-min data time-step, the flow increase applied to every step at the beginning of a rising limb was equal to $0.01843 m^3/s$, see Fig. B.1.

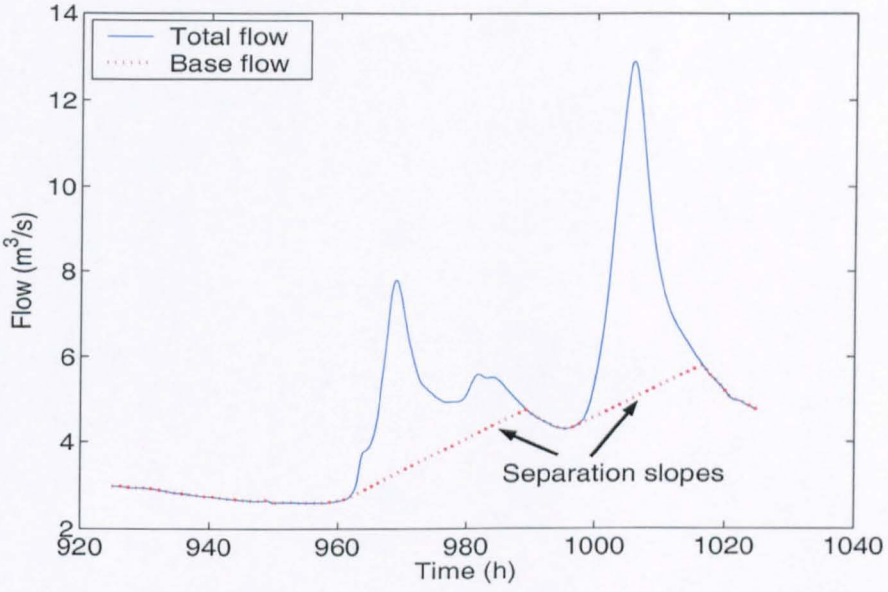


Figure B.1: Hydrograph separation into total and base flow.

B.3.2 The equation of the transfer function model

The model has been defined by a transfer function with three poles, as suggested by Han (1991). In his work three unsatisfactory cases for hydrological applications were described and are shown in Fig. B.2.

According to the analysis carried out in his work, a transfer function applicable to hydrological purpose can be written as

$$y(t) = \sum_{i=1}^{N} a_i y(t-i) + \sum_{i=0}^{N} b_i x(t-i) \tag{B.8}$$

The coefficients a_i can be computed from

$$a_i = -C_N^{N-i}(-\beta)^{-i} \tag{B.9}$$

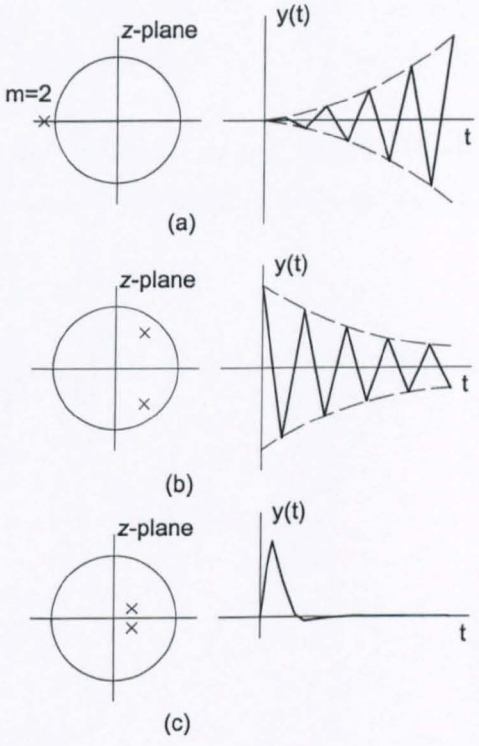


Figure B.2: Time-domain behaviour of a transfer function as a function of the pole location: (a) unstable, (b) fluctuating, (c) negative response.

where $i = 1, 2, ..., N$ and

$$C_N^k = \frac{N(N-1)(N-2)...[N-(k-1)]}{k!} \tag{B.10}$$

$$\beta = e^{\frac{2t_{peak}+3}{t_{peak}^2+3t_{peak}+2}} \tag{B.11}$$

$$t_{peak} = C_t - M \tag{B.12}$$

The pole values are equal to $1/\beta$, N is the order pole, C_t is the estimated catchment response time and M is the number of $b_i x(t-1)$ parameters in Eq. B.8.

The coefficients b_i were obtained through a genetic algorithm search as suggested by Yang (1999). The equation employed to compare the ANN model was:

$$y(t) = a_1 y(t-1) - a_2 y(t-2) + a_3 y(t-3) + b_0 x(t) + b_1 x(t-1) + b_2 x(t-2) \quad \text{(B.13)}$$

where

- $a_1 = 2.6534$
- $a_2 = 2.3468$
- $a_3 = 2.3468$
- $b_0 = 0.058$
- $b_1 = 0.029$
- $b_2 = 0.029$

To obtain the best performance this equation was adjusted according to the three parameters developed by Han (1991):

Volume adjusting factor α

$$b_i' = (1+\alpha) b_i \quad \text{(B.14)}$$

Shape adjusting factor γ

$$a_i' = -C_N^{N-i}(-e^{\frac{1}{(\gamma+\frac{1}{ln\beta})}})^{-i} \tag{B.15}$$

And time adjusting factor τ

$$y(t) = \sum_{i=1}^{N} a_i y(t-i) + \sum_{i=0}^{N} b_i x(t-\tau-i) \tag{B.16}$$

The rainfall-runoff periods used to adjust the TF model were those used to train the ANN model.

Appendix C

Crosscorrelation

Crosscorrelation is a mathematical method applied to evaluate whether two functions, i.e. rainfall and runoff estimations, exhibit common features, and therefore have a certain level of correlation.

If a rainfall series is represented as $r(t)$ and a flow series is represented as $f(t)$, the *crosscorrelation* of $r(t)$ and $f(t)$ is a series $x_{rf}(l)$, which is defined as

$$x_{rf}(l) = \sum_{t=-\infty}^{\infty} r(t)f(t-l), \qquad l = 0, \pm 1, \pm 2, ... \tag{C.1}$$

or, equivalently, as

$$x_{rf}(l) = \sum_{t=-\infty}^{\infty} r(t+l)f(t), \qquad l = 0, \pm 1, \pm 2, ... \tag{C.2}$$

The index l is the time shift (or *lag*) parameter and the subscripts rf on the crosscorrelation series $x_{rf}(l)$ indicate the series being correlated.

In practice the crosscorrelation generates a series of values which indicate the level of correlation for each time shift. In the case of the rainfall and runoff series, the natural response of a catchment can be identified as the point of the highest value from the resulting series as shown in Fig. 3.18.

Index